

CONTEMPORARY MATHEMATICS

218

Domain Decomposition Methods 10

The Tenth International Conference
on Domain Decomposition Methods
August 10–14, 1997
Boulder, CO

Jan Mandel
Charbel Farhat
Xiao-Chuan Cai
Editors



American Mathematical Society
Providence, Rhode Island

Contents

Preface	xi
Part 1. Invited Presentations	1
Nonmatching Grids for Fluids	
YVES ACHDOU, GASSAN ABDOLAEV, JEAN-CLAUDE HONTAND, YURI A. KUZNETSOV, OLIVIER PIRONNEAU, AND CHRISTOPHE PRUD'HOMME	3
A Parallel Non-Overlapping Domain-Decomposition Algorithm for Compressible Fluid Flow Problems on Triangulated Domains TIMOTHY J. BARTH, TONY F. CHAN, AND WEI-PAI TANG	23
A Non-Overlapping Domain Decomposition Method for the Exterior Helmholtz Problem ARMEL DE LA BOURDONNAYE, CHARBEL FARHAT, ANTONINI MACEDO, FRÉDÉRIC MAGOULÈS, AND FRANÇOIS-XAVIER ROUX	42
An Agglomeration Multigrid Method for Unstructured Grids TONY F. CHAN, JINCHAO XU, AND LUDMIL ZIKATANOV	67
Solution of Coercive and Semicoercive Contact Problems by FETI Domain Decomposition ZDENĚK DOSTÁL, ANA FRIEDLANDER, AND SANDRA A. SANTOS	82
An Iterative Substructuring Method for Elliptic Mortar Finite Element Problems with Discontinuous Coefficients MAKSYMILIAN DRYJA	94
Domain Decomposition Methods for Flow in Heterogeneous Porous Media MAGNE S. ESPEDAL, KARL J. HERSVIK, AND BRIT G. ERSLAND	104
A Fictitious Domain Method with Distributed Lagrange Multipliers for the Numerical Simulation of Particulate Flow ROLAND GLOWINSKI, TSORNG-WHAY PAN, TODD I. HESLA, DANIEL D. JOSEPH, AND JACQUES PERIAUX	121
Domain Decomposition Algorithms for Saddle Point Problems LUCA F. PAVARINO	138
Parallel Implementation of Direct Solution Strategies for the Coarse Grid Solvers in 2-level FETI Method FRANÇOIS-XAVIER ROUX AND CHARBEL FARHAT	158
Domain Decomposition and Multi-Level Type Techniques for General Sparse Linear Systems YOUSEF SAAD, MARIA SOSONKINA, AND JUN ZHANG	174

Spectral/ <i>hp</i> Methods for Elliptic Problems on Hybrid Grids SPENCER J. SHERWIN, TIMOTHY C.E. WARBURTON, AND GEORGE EM KARNIADAKIS	191
Physical and Computational Domain Decompositions for Modeling Subsurface Flows MARY F. WHEELER AND IVAN YOTOV	217
Part 2. Algorithms	229
Nonoverlapping Domain Decomposition Algorithms for the <i>p</i> -version Finite Element Method for Elliptic Problems ION BICĂ	231
A 2-level and Mixed Domain Decomposition Approach for Structural Analysis DAVID DUREISSEIX AND PIERRE LADEVÈZE	238
Iso-P2 P1/P1/P1 Domain-Decomposition/Finite-Element Method for the Navier-Stokes Equations SHOICHI FUJIMA	246
Overlapping Nonmatching Grids Method: Some Preliminary Studies SERGE GOOSSENS, XIAO-CHUAN CAI, AND DIRK ROOSE	254
Nonconforming Grids for the Simulation of Fluid-Structure Interaction CÉLINE GRANDMONT AND YVON MADAY	262
Hash-Storage Techniques for Adaptive Multilevel Solvers and Their Domain Decomposition Parallelization MICHAEL GRIEBEL AND GERHARD ZUMBUSCH	271
Extension of a Coarse Grid Preconditioner to Non-symmetric Problems CAROLINE JAPHET, FRÉDÉRIC NATAF, AND FRANÇOIS-XAVIER ROUX	279
On the Interaction of Architecture and Algorithm in the Domain-based Parallelization of an Unstructured-grid Incompressible Flow Code DINESH K. KAUSHIK, DAVID E. KEYES, AND BARRY F. SMITH	287
Additive Domain Decomposition Algorithms for a Class of Mixed Finite Element Methods AXEL KLAWONN	296
Non-conforming Domain Decomposition Method for Plate and Shell Problems CATHERINE LACOUR	304
Solutions of Boundary Element Equations by a Flexible Elimination Process CHOI-HONG LAI AND KE CHEN	311
An Efficient FETI Implementation on Distributed Shared Memory Machines with Independent Numbers of Subdomains and Processors MICHEL LESOINNE AND KENDALL PIERSON	318
Additive Schwarz Methods with Nonreflecting Boundary Conditions for the Parallel Computation of Helmholtz Problems LOIS C. MCINNES, ROMEO F. SUSAN-RESIGA, DAVID E. KEYES, AND HAFIZ M. ATASSI	325

On the Reuse of Ritz Vectors for the Solution to Nonlinear Elasticity Problems by Domain Decomposition Methods FRANCK RISLER AND CHRISTIAN REY	334
Dual Schur Complement Method for Semi-Definite Problems DANIEL J. RIXEN	341
Two-level Algebraic Multigrid for the Helmholtz Problem PETR VANĚK, JAN MANDEL, AND MARIAN BREZINA	349
A Comparison of Scalability of Different Parallel Iterative Methods for Shallow Water Equations ARNT H. VEENSTRA, HAI XIANG LIN, AND EDWIN A.H. VOLLEBREGT	357
A Nonoverlapping Subdomain Algorithm with Lagrange Multipliers and Its Object Oriented Implementation for Interface Problems DAOQI YANG	365
Part 3. Theory	375
A Robin-Robin Preconditioner for an Advection-Diffusion Problem YVES ACHDOU AND FRÉDÉRIC NATAF	377
A Semi-dual Mode Synthesis Method for Plate Bending Vibrations FRÉDÉRIC BOURQUIN AND RABAH NAMAR	384
Overlapping Schwarz Algorithms for Solving Helmholtz's Equation XIAO-CHUAN CAI, MARIO A. CASARIN, FRANK W. ELLIOTT, JR., AND OLOF B. WIDLUND	391
Symmetrized Method with Optimized Second-Order Conditions for the Helmholtz Equation PHILIPPE CHEVALIER AND FRÉDÉRIC NATAF	400
Non-overlapping Schwarz Method for Systems of First Order Equations SÉBASTIEN CLERC	408
Interface Conditions and Non-overlapping Domain Decomposition Methods for a Fluid–Solid Interaction Problem XIAOBING FENG	417
Overlapping Schwarz Waveform Relaxation for Parabolic Problems MARTIN J. GANDER	425
Domain Decomposition, Operator Trigonometry, Robin Condition KARL GUSTAFSON	432
On Orthogonal Polynomial Bases for Triangles and Tetrahedra Invariant under the Symmetric Group GARY MAN-KWONG HUI AND HOWARD SWANN	438
On Schwarz Alternating Methods for Nonlinear Elliptic Problems SHIU HONG LUI	447
Convergence Results for Non-Conforming hp Methods: The Mortar Finite Element Method PADMANABHAN SESHAIYER AND MANIL SURI	453

Intergrid Transfer Operators for Biharmonic Problems Using Nonconforming Plate Elements on Nonnested Meshes ZHONGCI SHI AND ZHENGHUI XIE	460
Additive Schwarz Methods for Hyperbolic Equations YUNHAI WU, XIAO-CHUAN CAI, AND DAVID E. KEYES	468
Part 4. Applications	477
A Minimum Overlap Restricted Additive Schwarz Preconditioner and Applications in 3D Flow Simulations XIAO-CHUAN CAI, CHARBEL FARHAT, AND MARCUS SARKIS	479
Time Domain Decomposition for European Options in Financial Modelling DIANE CRANN, ALAN J. DAVIES, CHOI-HONG LAI, AND SWEE H. LEONG	486
Parallel Modal Synthesis Methods in Structural Dynamics JEAN-MICHEL CROS	492
Efficient Computation of Aerodynamic Noise GEORGI S. DJAMBAZOV, CHOI-HONG LAI, AND KOULIS A. PERICLEOUS	500
Non-overlapping Domain Decomposition Applied to Incompressible Flow Problems FRANK-CHRISTIAN OTTO AND GERT LUBE	507
A Domain Decomposition Based Algorithm for Non-linear 2D Inverse Heat Conduction Problems CHARAKA J. PALANSURIYA, CHOI-HONG LAI, CONSTANTINOS S. IEROTHEOU, AND KOULIS A. PERICLEOUS	515
Overlapping Domain Decomposition and Multigrid Methods for Inverse Problems XUE-CHENG TAI, JOHNNY FRØYEN, MAGNE S. ESPEDAL, AND TONY F. CHAN	523
Some Results on Schwarz Methods for a Low-Frequency Approximation of Time-Dependent Maxwell's Equations in Conductive Media ANDREA TOSELLI	530
Parallel Computing for Reacting Flows Using Adaptive Grid Refinement ROBBERT L. VERWEIJ, ARIS TWERDA, AND TIM W.J. PEETERS	538
The Coupling of Mixed and Conforming Finite Element Discretizations CHRISTIAN WIENERS AND BARBARA I. WOHLMUTH	547

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 ten 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 Tenth Conference on Domain Decomposition Methods took place at the University of Colorado at Boulder from August 10 to August 14, 1997. It was organized by Charbel Farhat, Department of Aerospace Engineering Science, Xiao-Chuan Cai, Department of Computer Science, both at the University of Colorado at Boulder, and Jan Mandel, Department of Mathematics at the University of Colorado at Denver.

Driven by the availability of powerful parallel processors, the field of Domain Decomposition has matured during the past ten years. The focus of new methods has been shifting from positive definite elliptic problems to complicated applications, nonlinear problems, systems, and problems with non-elliptic numerical behavior, such as wave propagation and the Helmholtz equation. At the same time, the advent of practical massively parallel computers poses new challenges for elliptic equations, especially on arbitrary, nonuniform meshes. These Proceedings contain contributions from all these areas. The focus of the Conference, as reflected in the selection of invited speakers, was on realistic applications in structural mechanics, structural dynamics, computational fluid dynamics, and heat transfer.

The Conference had 171 registered participants. There were 16 invited plenary lectures and 113 mini-symposia and plenary presentations. These proceedings contain 13 invited and 41 mini-symposia and contributed papers. All papers have been refereed. The Proceedings are divided into four parts. The first part contains invited papers. The rest of the volume contains mini-symposia and contributed presentations, further divided into Algorithms, Theory, and Applications.

Previous proceedings of the International Conferences on Domain Decomposition were published by SIAM, AMS, and John Wiley & Sons. We welcome the return of the Proceedings to AMS. We would like to acknowledge the help of the AMS staff in deciding the format and preparing the Proceedings. We would like to thank particularly Dr. Sergei Gelfand for encouraging us to abolish the page limit for invited presentations.

We wish to thank the members of the International Scientific Committee, and in particular the Chair, Petter Bjørstad, 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.

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 the National Science Foundation, ANSYS, Inc., the Sandia National Laboratories, the Colorado School of Mines, the University of Colorado at Boulder, and the University of Colorado at Denver. Their generous support made the Conference possible and, among other things, allowed the organizers to fund the participation of graduate students.

Finally, we would like to express our appreciation to Ms. Cathy Moser, the Secretary of the Conference, who made all organizational details run smoothly, and Dr. Radek Tezaur, the Technical Editor of these Proceedings, who finalized the formatting of the papers in AMS-L^AT_EX and prepared the whole book for printing.

The complete program of the Conference is available at the Conference Web site <http://www-math.cudenver.edu/dd10>. More related information, including links to other Domain Decomposition conferences and books, can be found at the Official Domain Decomposition Web site at <http://www.ddm.org>. The purchaser of this volume is entitled to the online edition of this book by AMS. To gain access, follow the instructions given on the form found in the back of this volume.

Jan Mandel
Charbel Farhat
Xiao-Chuan Cai

Part 1

Invited Presentations

Nonmatching Grids for Fluids

Yves Achdou, Gassan Abdoulaev, Jean-Claude Hontand,
Yuri A. Kuznetsov, Olivier Pironneau, and Christophe Prud'homme

1. Introduction

We review some topics about the use of nonmatching grids for fluids. In a first part, we discuss the mortar method for a convection diffusion equation. In a second part, we present a three dimensional Navier-Stokes code, based on mortar elements, whose main ingredients are the method of characteristics for convection, and a fast solver for elliptic equations for incompressibility. Finally, preliminary numerical results are given.

Since the late nineteen eighties, interest has developed in non-overlapping domain decomposition methods coupling different variational approximations in different subdomains. The *mortar element methods*, see [10], [28], have been designed for this purpose and they allow us to combine different discretizations in an optimal way. Optimality means that the error is bounded by the sum of the subregion-by-subregion approximation errors without any constraints on the choice of the different discretizations. One can, for example couple spectral methods with finite elements, or different finite element methods with different meshes.

The advantages of the mortar method are:

- It permits to use spectral methods with flexibility.
- It is well suited for partial differential equations involving highly heterogeneous media.
- Mesh adaption can be made local [9].
- It enables one to assemble several meshes generated independently: this is important, since constructing meshes is not an easy problem in three dimensions.
- In the finite elements context, it permits the use of locally structured grids in the subdomains, thence fast local solvers. Besides, for computational fluid dynamics, structured grids are very helpful in boundary layers for instance.
- It permits the use of sliding meshes see [6].

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 76D05, 65M60, 65N55.

The second and fourth authors were partially supported by French-Russian Liapunov Institute for Informatics and Applied Mathematics and INRIA. This work was partially carried out at the CEMRACS summer school, <http://www.asci.fr/cemracs>. The first author was partially supported by CNRS PICS 478. The computations were performed on the Cray T3E at IDRIS, Orsay, France.

In this paper, we focus on the use of the mortar method for the simulation of fluid dynamics problems.

After reviewing the method for symmetric problems, we propose and analyze a mortar method suited for convection-diffusion problems. We show that upwinding terms should be added at the interfaces between subdomains.

Then, we discuss a 3D Navier-Stokes code with nonmatching grids, which is being developed at laboratory ASCI (Applications Scientifiques du Calcul Intensif) in Orsay(France). This code is based on a projection-characteristics scheme [4] and on a fast parallel solver for the pressure. With such a scheme, the involved elliptic operators are time-invariant and symmetric. The solver we use for the pressure has been developed in [22] and tested in [1]. It is used for solving the saddle point linear system arising from the mortar element discretization of the pressure equation. Other solvers or preconditioners have been developed for nonmatching grid problems, e.g substructuring preconditioners in the space of weakly continuous functions [14, 5, 16], or Neumann-Neumann preconditioners [23].

Finally, we present some results both in 2D and 3D. The 2D results are also obtained with a mortar element method, but with a vorticity stream-function formulation. The 3D results are preliminary, since the code is in progress.

2. A brief review on the mortar method

Assume that we have to solve the elliptic problem: find $u \in H^1(\Omega)$ such that

$$(1) \quad \tilde{a}(u, v) = (f, v), \quad \forall v \in H^1(\Omega)$$

in a two dimensional domain Ω , where $\tilde{a}(\cdot, \cdot)$ is a second order bilinear form, continuous on $H^1(\Omega)$. We shall assume for simplicity that $\tilde{a}(u, v) = \int_{\Omega} \nabla u \cdot \nabla v$, which corresponds to a Neumann problem, and that the right hand side f is such that (1) has a solution, unique up to the addition of constants. Consider a nonoverlapping domain decomposition $\bar{\Omega} = \bigcup_{k=1}^K \bar{\Omega}_k$. We assume that each subdomain Ω_k is a piecewise regular, curved polygon. For simplicity only we also suppose that the domain decomposition is geometrically conforming. It means that if $\gamma_{kl} = \bar{\Omega}_k \cap \bar{\Omega}_l$ ($k \neq l$) and $\gamma_{kl} \neq \emptyset$, then γ_{kl} can be either a common vertex of Ω_k and Ω_l , or a common edge, or a common face. In the last case we define $\Gamma_{kl} = \gamma_{kl}$ as the interface between Ω_k and Ω_l . Note that $\Gamma_{kl} = \Gamma_{lk}$. The union Γ of all $\partial\Omega_k \setminus \partial\Omega$ is called the skeleton (or interface) of the decomposition.

Let us introduce the spaces $X_k \equiv H^1(\Omega_k)$. It is also possible to define the local bilinear forms a_k on $X_k \times X_k$ by $a_k(u_k, v_k) = \int_{\Omega_k} \nabla u_k \cdot \nabla v_k$, and the global bilinear form

$$(2) \quad a : \prod_{k=1}^K X_k \times \prod_{k=1}^K X_k \rightarrow \mathbb{R}, \quad a(u, v) = \sum_{k=1}^K a_k(u_k, v_k).$$

Suppose that each Ω_k is provided with its own discretization $X_{k,h}$ of X_k . In what follows, the discrete spaces are of finite element type, but the proposed method can also be used for any discretization of the variational problem. To derive a global discretization of $H^1(\Omega)$, yielding an accurate Galerkin approximation of (1), the mortar element method was introduced in [10]. Reviewing the basics of the method, the global discrete space is a subspace of $X_h \equiv \prod_{1 \leq k \leq K} X_{k,h}$ obtained by imposing matching conditions on each face Γ_{kl} . On any such face, there are two

values, the traces v_k (respectively v_l) of a function defined in Ω_k (respectively Ω_l). For $k < l$ such that $|\partial\Omega_l \cap \partial\Omega_k| > 0$, denote by $\tilde{W}_{k,l,h}$ (respectively $\tilde{W}_{l,k,h}$) the trace of the finite element space $X_{k,h}$ (respectively $X_{l,h}$) on Γ_{kl} . Let $W_{k,l,h}$ be a subspace of codimension two of either $\tilde{W}_{k,l,h}$ or $\tilde{W}_{l,k,h}$. We refer to [10] for more details and the exact definition of this subspace. We then set

$$(3) \quad Y_h \equiv \left\{ \begin{array}{l} \mathbf{v} \in X_h : \text{for all } k < l \text{ such that } |\partial\Omega_l \cap \partial\Omega_k| > 0, \\ \int_{\Gamma_{kl}} (v_k - v_l)\psi = 0, \quad \forall \psi \in W_{k,l,h} \end{array} \right\}.$$

The discrete problem reads: find $u_h \in Y_h$ such that

$$(4) \quad \forall v_h \in Y_h, \quad a(u_h, v_h) = \int_{\Omega} fv_h.$$

It has been established in [10] that the resulting discretization error is as good as for comparable conforming finite element or spectral approximations. Indeed, if the solution of (1) is regular enough (in $H^{\frac{3}{2}+\epsilon}$) then the error in the norm $\|v\|_{1,*} \equiv \sqrt{a(v,v)}$ is bounded by a constant times the sum of the subregion by subregion best approximation errors.

3. The mortar method for convection-diffusion problems

In this section, we present a mortar method suited for non symmetric problems. Since this method is only studied theoretically here and is not implemented in our Navier-Stokes code, this section can be seen as a digression. Let us consider the stationary scalar linear convection dominated convection-diffusion equation:

$$(5) \quad \begin{aligned} -\epsilon\Delta u + \nabla \cdot (\beta u) + \sigma u &= f && \text{in } \Omega, \\ u &= g && \text{on } \partial\Omega, \end{aligned}$$

where Ω is a bounded polygonal domain in \mathbb{R}^2 , σ and ϵ are two constants, ϵ being small. It is assumed that the velocity β is smooth and that $\sigma - \frac{1}{2}|\nabla \cdot \beta| > \bar{\sigma} > 0$. In the following, we take $g = 0$ in (5).

If optimal error estimates are desired, the mortar method reviewed above cannot be applied in a straightforward manner. We shall show that upwinding terms are needed at the interfaces between subdomains.

In this section, we work with triangular meshes in the subdomains and P_1 elements and we assume that the triangulations in the subdomains are regular and quasi uniform. Let h_k be the maximal diameter of the triangles in Ω_k . We shall assume that $\epsilon \ll h_k$. In this case, introducing upwinding by using e.g. streamline diffusion methods, see [20], is necessary because a standard Galerkin approach for discretizing (5) would produce oscillations.

We denote by X_{kh} the space of continuous and piecewise linear functions on the triangulation of Ω_k , vanishing on $\partial\Omega_k \cap \partial\Omega$, and Y_h is defined as in (3).

Consider a family $(\delta_k)_{1 \leq k \leq K}$ of small positive parameters : $\delta_k \sim h_k$ and the non symmetric bilinear form $a_k : X_k \times X_k \rightarrow \mathbb{R}$:

$$(6) \quad a_k(u_k, v_k) = \epsilon \int_{\Omega_k} \nabla u_k \cdot \nabla v_k + \int_{\Omega_k} (\sigma u_k + \nabla \cdot (\beta u_k))(v_k + \delta_k \nabla \cdot (\beta v_k)).$$

It corresponds to a streamline diffusion method applied in subdomain Ω_k , see [20]. However, the discrete problem: find $u_h \in Y_h$ such that

$$(7) \quad \forall v_h \in Y_h, \quad a(u_h, v_h) = \sum_{k=1}^K \int_{\Omega_k} f(v_{kh} + \delta_k \nabla \cdot (\beta v_{kh}))$$

with

$$(8) \quad a(u, v) = \sum_{k=1}^K a_k(u_k, v_k)$$

would not necessarily yield optimal error estimates. Indeed,

$$(9) \quad \begin{aligned} a(u, u) = & \\ & \sum_{1 \leq k \leq K} \int_{\Omega_k} \epsilon |\nabla u_k|^2 + (\sigma + \frac{1}{2} \nabla \cdot \beta) u_k^2 + h_k |\nabla \cdot (\beta u_k)|^2 + \sigma h_k \nabla \cdot (\beta u_k) u_k \\ & + \frac{1}{2} \sum_{|\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} \beta \cdot n_{kl} (u_k^2 - u_l^2) \end{aligned}$$

and it is not possible to control the term

$$\frac{1}{2} \sum_{|\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} \beta \cdot n_{kl} (u_k^2 - u_l^2)$$

in an optimal way. Therefore, we shall also introduce a stabilizing term for each interface Γ_{kl} : for $k < l : |\Gamma_{kl}| \neq 0$, let us define the bilinear form $\tilde{a}_{kl} : (X_k \times X_l)^2 \rightarrow \mathbb{R}$:

$$(10) \quad \tilde{a}_{kl}(u_k, u_l, v_k, v_l) = \int_{\Gamma_{kl}} (\beta \cdot n_{kl})^- (u_k - u_l) v_k + \int_{\Gamma_{kl}} (\beta \cdot n_{lk})^- (u_l - u_k) v_l,$$

where x^- denotes the negative part of x : $x^- = \frac{1}{2}(|x| - x)$. The bilinear form $a : X \times X \rightarrow \mathbb{R}$ is now defined by

$$(11) \quad a(u, v) = \sum_{1 \leq k \leq K} a_k(u_k, v_k) + \sum_{k < l : |\Gamma_{kl}| \neq 0} \tilde{a}_{kl}(u_k, u_l, v_k, v_l).$$

The chosen discretization of (5) is: find $u_h \in Y_h$ such that

$$(12) \quad \forall v_h \in Y_h, \quad a(u_h, v_h) = \sum_{1 \leq k \leq K} \int_{\Omega_k} f(v_{kh} + \delta_k \nabla \cdot (\beta v_{kh})).$$

REMARK 1. Note that the upwinding is done by the streamline diffusion method inside the subdomains, and by some discontinuous Galerkin like method at the interfaces between subdomains, where no strong continuity is available.

Let $\|v\|$ denotes the broken norm :

$$(13) \quad \|v\|^2 \equiv \sum_{1 \leq k \leq K} \left\{ \epsilon \int_{\Omega_k} |\nabla v_k|^2 + \frac{\bar{\sigma}}{2} \int_{\Omega_k} v_k^2 + \frac{\delta_k}{2} \int_{\Omega_k} (\nabla \cdot (\beta v_k))^2 \right\},$$

and $|v|$ the semi-norm

$$(14) \quad |v|^2 \equiv \frac{1}{2} \sum_{k < l : |\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} |\beta \cdot n_{kl}| (v_k - v_l)^2$$

We will prove an error estimate in the following norm :

$$(15) \quad |||v||| \equiv (\|v\|^2 + |v|^2)^{\frac{1}{2}}.$$

The stability of the method stems from the following lemma:

LEMMA 2. If $\forall k \in \{1, K\}$, $\delta_k \leq \frac{\bar{\sigma}}{2\sigma^2}$, then for any $v \in Y_h$,

$$(16) \quad a(v, v) \geq |||v|||^2.$$

From this lemma and from the best fit estimate in the L^2 -norm when approaching a sufficiently regular function by a function in Y_h (see [8]), we obtain the following result, which states that optimal error estimates are obtained, similarly as for the conforming case.

PROPOSITION 3. Assume that the solution of (5) is in $H_0^1(\Omega) \cap \prod_{1 \leq k \leq K} H^2(\Omega_k)$

and that

$$(17) \quad \forall k \in \{1, K\}, \quad \epsilon \leq ch_k \quad \text{and} \quad \delta_k = h_k,$$

then there exists a constant C such that, if u_h is the solution of (12),

$$(18) \quad |||u - u_h||| \leq C \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)}.$$

PROOF. It is seen from [10], [8] (see in particular [8], remark 7), that there exists \tilde{u}_h in Y_h such that $w_h = u - \tilde{u}_h$ satisfies

$$(19) \quad \left(\sum_{1 \leq k \leq K} \int_{\Omega_k} |\nabla w_{kh}|^2 \right)^{\frac{1}{2}} \leq C \sum_{1 \leq k \leq K} h_k \|u_k\|_{H^2(\Omega_k)}$$

and

$$(20) \quad \left(\sum_{1 \leq k \leq K} \int_{\Omega_k} w_{kh}^2 \right)^{\frac{1}{2}} \leq C \sum_{1 \leq k \leq K} h_k^2 \|u_k\|_{H^2(\Omega_k)}.$$

These approximations results have been obtained when proving that the mortar method leads to optimal error estimates for an elliptic problem. Choosing other jump condition on interface, i.e. other spaces Y_h often lead to poorer approximation results (see [10]). Therefore, from assumption (17) ,

$$(21) \quad |||w_h||| \leq C \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)}.$$

Let $\tilde{e}_h = \tilde{u}_h - u_h$ and $e_h = u - u_h$. From (5) and (12) one obtains that

$$(22) \quad a(e_h, \tilde{e}_h) = \epsilon \sum_{1 \leq k \leq K} \delta_k \int_{\Omega_k} \Delta u \nabla \cdot (\beta \tilde{e}_{kh}) + \epsilon \sum_{k < l : |\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} \frac{\partial u}{\partial n_{kl}} (\tilde{e}_{kh} - \tilde{e}_{lh}).$$

Therefore since $\tilde{e}_h = e_h - w_h$,

$$(23) \quad \begin{aligned} & |||e_h|||^2 \\ &= a(e_h, w_h) + \\ & \epsilon \sum_{1 \leq k \leq K} \delta_k \int_{\Omega_k} \Delta u \nabla \cdot (\beta(e_{kh} - w_{kh})) + \epsilon \sum_{k < l : |\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} \frac{\partial u}{\partial n_{kl}} (\tilde{e}_{kh} - \tilde{e}_{lh}). \end{aligned}$$

The last two terms are consistency errors. Using arguments very similar to those of [20], it can be proved that

$$(24) \quad |a(e_h, w_h)| \leq C |||e_h||| \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)}.$$

It can also be seen that

$$\begin{aligned}
(25) \quad & \epsilon \left| \sum_{1 \leq k \leq K} \delta_k \int_{\Omega_k} \Delta u \nabla \cdot (\beta(e_{kh} - w_{kh})) \right| \\
& \leq \epsilon \|\Delta u\|_{L^2(\Omega)} \sum_{1 \leq k \leq K} \delta_k \|\nabla \cdot (\beta(e_{kh}\|_{L^2(\Omega_k)} + \delta_k \|\nabla \cdot (\beta(w_{kh}\|_{L^2(\Omega_k)} \\
& \leq C \left(|||e_h||| + \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)} \right) \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)}.
\end{aligned}$$

Finally, since $\tilde{e} \in Y_h$, calling π_h the L^2 projection on W_h (see [10]),

$$(26) \quad \epsilon \left| \int_{\Gamma_{kl}} \frac{\partial u}{\partial n_{kl}} (\tilde{e}_{kh} - \tilde{e}_{lh}) \right| = \epsilon \left| \int_{\Gamma_{kl}} (I - \pi_h) \frac{\partial u}{\partial n_{kl}} (\tilde{e}_{kh} - \tilde{e}_{lh}) \right|,$$

and using the lemma 4.1 of [10],

$$\begin{aligned}
(27) \quad & \epsilon \left| \sum_{k < l : |\Gamma_{kl}| \neq 0} \int_{\Gamma_{kl}} \frac{\partial u}{\partial n_{kl}} (\tilde{e}_{kh} - \tilde{e}_{lh}) \right| \\
& \leq C \left(|||e_h||| + \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)} \right) \sum_{1 \leq k \leq K} h_k^{\frac{3}{2}} \|u_k\|_{H^2(\Omega_k)}.
\end{aligned}$$

Collecting all the above estimates yields the desired result. \square

Such a discretization can be generalized to nonlinear conservation laws, see [3]:

$$\begin{aligned}
(28) \quad & -\epsilon \Delta u + \nabla \cdot f(u) + \sigma u = d & \text{in } \Omega \\
& u = 0 & \text{on } \partial\Omega,
\end{aligned}$$

As above, one must add a flux term on the interfaces

$$(29) \quad a_{kl}(u_k, u_l, v_k, v_l) = - \int_{\Gamma_{kl}} n_{kl} \cdot (f(u_k)v_k - f(u_l)v_l) + \int_{\Gamma_{kl}} n_{kl} \cdot g(u_k, u_l)(v_k - v_l)$$

where $n_{kl} \cdot (g(u_k, u_l)$ is a Godunov flux:

$$n_{kl} \cdot g(u_k, u_l) = \min_{u_k \leq u \leq u_l} n_{kl} \cdot f(u) \quad \text{if } u_k \leq u_l.$$

One can choose instead an Osher flux:

$$n_{kl} \cdot g(u_k, u_l) = \frac{1}{2} n_{kl} \cdot (f(u_k) + f(u_l)) + \frac{1}{2} \int_{(u_k, u_l)} |n_{kl} \cdot f'(u)|.$$

4. A Navier Stokes solver using the mortar element method

Getting back to what we have implemented, the proposed scheme is based on both projection and the characteristic Galerkin methods. As we shall see, the main advantage of such a scheme is that

- Thanks to the characteristics methods, the linear problems solved at each time step involve time invariant and symmetric elliptic operators.
- Thanks to the projection method, the problems for pressure and velocities are decoupled since the linearized Stokes problem is not solved exactly.

The code is implemented in three dimensions and is based on a fast solver for the pressure, which shall also be reviewed in this section. We shall also present 2D results with a mortar method, but the 2D code is based on a stream function-vorticity formulation.

In the following, we consider the time-dependent Navier-Stokes problem in which homogeneous Dirichlet condition has been assumed for simplicity. For a given body force f (possibly dependent on time) and a given divergence-free initial velocity field u_0 , find a velocity field u and a pressure field p such that $u = u_0$ at $t = 0$, and for $t > 0$,

$$(30) \quad \begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u + \nabla p &= f && \text{in } \Omega \times (0, T), \\ \nabla \cdot u &= 0 && \text{in } \Omega \times (0, T), \\ u &= 0 && \text{on } \partial\Omega \times (0, T). \end{aligned}$$

4.1. The scheme.

4.1.1. *The Galerkin characteristics method.* As in [25],[27], the total time derivatives are approximated by a finite difference formula in space and time, leading to a Eulerian-Lagrangian method: $\frac{Du}{Dt} = \frac{\partial u}{\partial t} + u \cdot \nabla u$ is discretized by

$$\frac{1}{\delta t}(u^{m+1} - u^m \circ X^m),$$

where $X^m(x)$ is the solution at time t^m of the backward Cauchy problem:

$$(31) \quad \frac{dX^m}{d\tau}(x, \tau) = u^m(X^m(x, \tau)) \quad \tau \in [t^m, t^{m+1}],$$

$$(32) \quad X^m(x, t^{m+1}) = x.$$

It can be seen easily that

$$w^m \circ X^m(x, t^m) \approx w^m(x - \delta t u^m(x)).$$

Therefore, at each time step, if we did not combine the characteristics method with a projection scheme, we would have to solve the linearized Stokes problem

$$(33) \quad \frac{1}{\delta t}(u^{m+1} - u^m \circ X^m(x, t^m)) - \nu \delta u^{m+1} + \nabla p^{n+1} = 0, \quad \text{in } \Omega$$

$$(34) \quad \nabla \cdot u^{n+1} = 0 \quad \text{in } \Omega.$$

4.1.2. *The spatial discretization.* We introduce \tilde{V}_h and Y_h two mortar finite element approximations of $(H_0^1(\Omega))^3$ and $L^2(\Omega)$. For instance, we can take a mortar version of the Q_1 iso Q_2 , Q_1 method: if the subdomains Ω_k are meshed with hexahedrons, and if the functions of $X_{k,h}$ are continuous and piecewise polynomial of degree 1 in each variable (Q_1), the space Y_h is given by (3). Then the mortar space \tilde{V}_h is composed of the Q_1 functions on the finer mesh (obtained by dividing each hexahedron of the initial mesh into eight hexahedra), vanishing on $\partial\Omega$ and matched at subdomains interfaces with a corresponding mortar condition.

We assume that the two spaces satisfy the Babuška-Brezzi inf-sup condition [12, 7]: there exists $c > 0$ such that

$$\inf_{q_h \in Y_h} \sup_{v_h \in \tilde{V}_h} \frac{\sum_{k=1}^K \int_{\Omega_k} v_{kh} \cdot \nabla q_{kh}}{\|v_h\|_{1,*} \|q_h\|_0} \geq c.$$

We now introduce a discrete divergence operator $C_h : \tilde{V}_h \rightarrow Y_h$ and its transpose $C_h^T : Y_h \rightarrow \tilde{V}_h'$ as follows: for every couple (v_h, q_h) in $\tilde{V}_h \times Y_h$ we have

$$(35) \quad (C_h v_h, q_h) = \sum_{k=1}^K \int_{\Omega_k} v_{kh} \cdot \nabla q_{kh} = (v_h, C_h^T q_h).$$

Calling $A_h : \tilde{V}_h \rightarrow \tilde{V}_h'$ the stiffness operator for the velocity

$$(A_h v_h, w_h) = \frac{1}{\delta t} \sum_{k=1}^K \int_{\Omega_k} v_{kh} w_{kh} + \nu \sum_{k=1}^K \int_{\Omega_k} \nabla v_{kh} \cdot \nabla w_{kh},$$

the discrete Stokes problem at time t^{m+1} reads:

$$(36) \quad A_h v_h^{m+1} + C_h^T q_h^{m+1} = g^{m+1}$$

$$(37) \quad C_h v_h^{m+1} = 0,$$

where g^{m+1} is computed from $u^m \circ X^m(x, t^m)$.

4.1.3. The characteristic-projection scheme. The projection method was introduced in [15] and [29]. It consists of using two approximations for the velocity at time step t^{m+1} namely u_h^{m+1} and \tilde{u}_h^{m+1} . The approximation \tilde{u}_h^{m+1} is sought in the previously defined space \tilde{V}_h . The second approximation u_h^{m+1} belongs to the space

$$(38) \quad V_h \equiv \tilde{V}_h + \tilde{\nabla} Y_h,$$

where the operator $\tilde{\nabla}$ is defined by:

$$(39) \quad \tilde{\nabla} : Y_h \rightarrow \prod_{k=1}^K (L^2(\Omega_k))^3$$

$$(40) \quad \tilde{\nabla} v_h = (\nabla v_{1h}, \dots, \nabla v_{Kh}).$$

It is possible to extend the operator C_h by $D_h : D_h : V_h \rightarrow Y_h$:

$$(41) \quad (D_h v_h, q_h) = \sum_{k=1}^K \int_{\Omega_k} v_{kh} \cdot \nabla q_{kh}$$

and to define D_h^T by $(v_h, D_h^T q_h) = (D_h v_h, q_h)$.

We are now interested in defining a projection/Lagrange-Galerkin scheme. We define two sequences of approximate velocities $\{\tilde{u}_h^m \in \tilde{V}_h\}$ and $\{u_h^m \in V_h\}$ and one sequence of approximate pressures $\{p_h^m \in Y_h\}$ as follows:

- **Initialization:** the sequences $\{u_h^m\}$, $\{\tilde{u}_h^m\}$ are initialized by for example $u_h^0 = \tilde{u}_h^0 = \hat{u}_h^0$ and the sequence $\{p_h^m\}$ is initialized by $p_h^0 = \hat{p}_h^0$.
- **Time loop:** For $0 \leq m$, solve

$$(42) \quad \begin{aligned} & (A_h \tilde{u}_h^{m+1}, v_h) - \left(\frac{u_h^m - \tilde{u}_h^m(X_h^m)}{\delta t}, v_h \right) + \left(\frac{\tilde{u}_h^m - \tilde{u}_h^m(X_h^m)}{\delta t}, v_h \right) = (f(t^{m+1}), v_h), \\ & \forall v_h \in \tilde{V}_h, \end{aligned}$$

and

$$(43) \quad \begin{aligned} & \frac{u_h^{m+1} - \tilde{u}_h^{m+1}}{\delta t} + D_h^T p_h^{m+1} = 0, \\ & D_h u_h^{m+1} = 0. \end{aligned}$$

The velocity u_h^{m+1} can be seen as the L^2 projection of \tilde{u}_h^{m+1} on the space of discrete divergence free functions. It satisfies only a weak non penetration condition at the boundary.

In practice, the projected velocity u_h^m must be eliminated from the algorithm as follows (see Rannacher [26] or Guermond [19]). For $m \geq 1$, replace u_h^m in (42) by its definition which is given by (43) at the time step t^m . In (43), u_h^{m+1} is eliminated by applying D_h to the first equation and by noting that D_h is an extension of C_h . If by convention we set $p_h^{-1} = \hat{p}_h^0$, the algorithm which should be implemented in practice reads for $m \geq 0$:

$$(44) \quad (A\tilde{u}_h^{m+1}, v_h) + (C_h^T p_h^m, v_h) = (f(t^{m+1}), v_h) + \left(\frac{\tilde{u}_h^m(X_h^m)}{\delta t}, v_h \right) \quad \forall v_h \in \tilde{V}_h,$$

and

$$(45) \quad (D_h D_h^T p_h^{m+1}, q_h) = \left(\frac{C_h \tilde{u}_h^{m+1}}{\delta t}, q_h \right) \quad \forall q_h \in Y_h.$$

Thanks to the choice of the space V_h (38) and of discretization for the divergence operator (41), the projection step can be rewritten: find p_h^{m+1} in Y_h such that

$$(46) \quad \forall q_h \in Y_h, \quad \sum_{k=1}^K \int_{\Omega_k} \nabla p_{kh}^{m+1} \cdot \nabla q_{kh} = \frac{1}{\delta t} \sum_{k=1}^K \int_{\Omega_k} \tilde{u}_{kh}^{m+1} \cdot \nabla q_{kh},$$

which is exactly the mortar discretization of the Poisson problem (1).

This algorithm has been analyzed for conforming discretizations (no mortars) in [4]. Roughly speaking, the results are the following: It is shown that provided the time step is of $\mathcal{O}(h^{d/4})$, where h is the mesh size and d is the space dimension ($2 \leq d \leq 3$), the proposed method yields for finite time T an error of $\mathcal{O}(h^{l+1} + \delta t)$ in the L^2 norm for the velocity and an error of $\mathcal{O}(h^l + \delta t)$ in the H^1 norm (or the L^2 norm for the pressure), where l is the spatial discretization order, if the solution is regular enough.

Second order schemes in time can also be implemented.

4.1.4. The projection method as a preconditioner. As shown by Cahouet and Chabard [13], Guermond[19], or Turek [30], the ingredients used by the projection method can be used to construct a good preconditioner for the generalized Stokes problem. Therefore, it is possible to solve iteratively the Stokes problem (33) in the spaces $\tilde{V}_h \times Y_h$ instead of using a projection scheme. Such an approach would be efficient at high Reynolds numbers or with small time steps.

4.2. A preconditioner for the mortar saddle point problem in three dimensions. Here we work in three dimensions, and we suppose that the partitioning into subdomains is regular and quasiuniform. We define by d_k the diameter of Ω_k and d an average diameter: there exists two constants C_1 and C_2 such that $C_1 d < d_k < C_2 d, \forall k \in \{1, \dots, K\}$. We also suppose for simplicity that the meshes in the subdomains are also regular and quasiuniform with constants and mean diameter independent of the subdomain. In particular, there exists two constants c_1 and c_2 , such that for any $k \in \{1, \dots, K\}$ the diameter of the elements in Ω_k are greater than $c_1 h$ and smaller than $c_2 h$.

We consider the saddle point problem:

$$(47) \quad \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} V \\ \Lambda \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

arising from the discretization by the mortar method of the elliptic equation

$$(48) \quad -\Delta p + \epsilon p = f \quad \text{in } \Omega,$$

$$(49) \quad \frac{\partial p}{\partial n} = 0 \quad \text{on } \partial\Omega.$$

where ϵ is a positive parameter which may be taken as small as desired. Thus the Neumann problems in the subdomains are well posed. We use this equation as an approximation of the Poisson problem that is satisfied by the pressure, with $\epsilon = 10^{-4}$ or $\epsilon = 10^{-5}$.

The matrix A is a block diagonal matrix, each block corresponds to a discrete Neumann problem in Ω_k . The matrix B is the matrix of the jump bilinear form b_h in the nodal bases of X_h and $W_h \equiv \prod_{|\partial\Omega_l \cap \partial\Omega_k| > 0} W_{k,l,h}$:

$$(50) \quad b_h(u_h, \mu_h) \equiv \sum_{k < l : |\partial\Omega_l \cap \partial\Omega_k| > 0} \int_{\Gamma_{kl}} (u_{kh} - u_{lh}) \mu_{kth}$$

As proved in [8], the saddle point problem (47) enjoys a Babuška-Brezzi condition with an inf-sup constant independent on the parameter h . Therefore, (47) is well posed.

The preconditioned iterative algorithm described below has been proposed by Y. Kuznetsov in [22] and its parallel implementation has been fully described and tested in [1]. We assume that the subdomains have an aspect ratio bounded by a constant and that the coarse mesh is quasi uniform. Let \mathcal{A} be a symmetric invertible matrix and let \mathcal{B} be a symmetric and positive definite matrix. If the solutions of the generalized eigenvalue problem

$$\mathcal{A}X = \nu \mathcal{B}X$$

belong to the union of the segments $[d_1; d_2] \cup [d_3; d_4]$, where $d_1 \leq d_2 < 0 < d_3 \leq d_4$, the generalized Lanczos method of minimal iterations [24] for solving iteratively the system $\mathcal{A}X = F$, with the preconditioner \mathcal{B} has a convergence rate depending on the generalized condition number $\kappa = \frac{\max\{d_4, |d_1|\}}{\min\{d_3, |d_2|\}}$.

Therefore, the idea is to use as a preconditioner for the saddle point problem (47) the matrix

$$\mathcal{B} = \begin{pmatrix} R_u & 0 \\ 0 & R_\lambda \end{pmatrix},$$

where R_u (resp. R_λ) is spectrally equivalent to A , (resp. $S_\lambda = BA^{-1}B^T$) (by spectrally equivalent, we mean that the condition number does depend neither on h , nor on the diameters of the subdomains nor on ϵ). For such a choice of \mathcal{B} , it can be proved that the generalized condition number κ does not depend on the parameters above. The matrices R_u and R_λ must also be chosen such that the systems $R_u v = w$ and $R_\lambda \mu = \beta$ are easy and inexpensive to solve.

It is natural to take R_u block diagonal with one block per subdomain: $R_u = \text{diag}(R_k)$ with the blocks R_k spectrally equivalent to the matrices A_k . Denote

$$(51) \quad \widehat{A}_k = \overset{\circ}{A}_k + \frac{1}{d^2} M_k,$$

where $\overset{\circ}{A}_k$ is the stiffness matrix (related to the Laplace operator with Neumann boundary condition in Ω_k), M_k is the mass matrix. Let P_k be the l_2 orthogonal projector onto the kernel of $\overset{\circ}{A}_k$, $\widehat{P}_k = I_k - P_k$ and H_k is any symmetric positive definite matrix spectrally equivalent to $\overset{\circ}{A}_k^{-1}$. Then, if R_k is chosen such that

$$(52) \quad R_k^{-1} = \widehat{P}_k H_k \widehat{P}_k + \frac{1}{\epsilon h^3} P_k,$$

it can be proved that R_k is spectrally equivalent to A_k [22]. It is for example possible to choose H_k as an additive multilevel preconditioner (see [11, 31, 18]), and this proves very efficient, since the cost of the matrix-vector product is proportional to the number of unknowns, and therefore optimal. The choice of the exponent 3 in the formula (52) is linked to the dimension and would have to be changed in dimension 2.

The construction of the preconditioner for S_λ consists of three stages:

- the first step consists of constructing a matrix \widehat{S}_λ spectrally equivalent to S_λ such that the product of a vector by \widehat{S}_λ is rather cheap;
- in the second step, a coarse grid preconditioner \widehat{R}_λ is proposed for S_λ or equivalently for \widehat{S}_λ ;
- in the third step, a generalized Chebyshev method with preconditioner \widehat{R}_λ is applied to the matrix \widehat{S}_λ .

Let us call S_k the discrete Dirichlet to Neumann operator (Schur complement of the matrix A_k). The matrix \widehat{S}_λ is chosen in the form

$$(53) \quad \widehat{S}_\lambda = \sum_{k=1}^K B_k \widetilde{H}_k B_k^T,$$

where B_k is the block of B corresponding with the degrees of freedom located on $\partial\Omega_k$, and where \widetilde{H}_k is spectrally equivalent to S_k^{-1} : the matrix \widetilde{H}_k is chosen as

$$(54) \quad \widetilde{H}_k = \widehat{P}_{\Gamma_k} \widehat{H}_{\Gamma_k} \widehat{P}_{\Gamma_k} + \frac{1}{\epsilon d h^2} P_{\Gamma_k},$$

where P_{Γ_k} is the l_2 orthogonal projector onto the one dimensional space spanned by the vector of dimension n_{Γ_k} (n_{Γ_k} is the number of grid nodes on $\partial\Omega_k$) whose components are 1, $\widehat{P}_{\Gamma_k} = I_{\Gamma_k} - P_{\Gamma_k}$. The matrix \widehat{H}_{Γ_k} is chosen to be spectrally equivalent to

$$\left(\overset{\circ}{S}_{\Gamma_k} + \frac{1}{d} M_{\Gamma_k} \right)^{-1},$$

where M_{Γ_k} denotes the $n_{\Gamma_k} \times n_{\Gamma_k}$ mass matrix on $\partial\Omega_k$ and where $\overset{\circ}{S}_{\Gamma_k}$ is the Schur complement of the matrix $\overset{\circ}{A}_k$.

The coarse space preconditioner \widehat{R}_λ is defined as

$$(55) \quad \widehat{R}_\lambda = \sum_{k=1}^K D_k + \alpha B_{\Gamma_k} P_{\Gamma_k} {B_{\Gamma_k}}^T, \quad \alpha = \frac{1}{\epsilon d h^2},$$

where D_k is a diagonal matrix, obtained by lumping the matrix $(1/h)B_{\Gamma_k}B_{\Gamma_k}^T$

It is proved in [22] that the condition number of $\widehat{R}_\lambda^{-1}\widehat{S}_\lambda$ can be bounded by cd/h where c is a positive constant independent of h , d and the small parameter ϵ .

Finally, the inverse of the preconditioner R_λ can be defined by

$$(56) \quad R_\lambda^{-1} = \left[I_\lambda - \prod_{l=1}^L \left(I_\lambda - \beta_l \widehat{R}_\lambda^{-1} \widehat{S}_\lambda \right) \right] \widehat{S}_\lambda^{-1},$$

where I_λ is the $n_\lambda \times n_\lambda$ identity matrix (n_λ is the dimension of the space W_h), β_l are the Chebyshev parameters corresponding to the spectral bounds of $\widehat{R}_\lambda^{-1}\widehat{S}_\lambda$. If the number L of the preconditioned Chebyshev iterations is fixed to a value of the order $O(\sqrt{\frac{d}{h}})$, it can be proved that

$$R_\lambda \sim \widehat{S}_\lambda$$

thence,

$$R_\lambda \sim S_\lambda.$$

In order to have an optimal preconditioner in terms of arithmetical complexity, the matrix-vector product by \widehat{S}_λ should be done in $O(d^{-\frac{1}{2}}h^{-\frac{5}{2}})$ operations at most. It is possible to construct such a matrix by noticing that, if \widehat{H}_k is a matrix spectrally equivalent to the inverse of \widehat{A}_k , defined in (51), then the block of \widehat{H}_k , corresponding to the nodes located on the boundary of Ω_k , is spectrally equivalent to $(\widehat{S}_{\Gamma_k} + \frac{1}{d}M_{\Gamma_k})^{-1}$. For example, if \widehat{S}_k is defined by using the Bramble Paschiak and Xu preconditioner [11] or the multilevel diagonal scaling preconditioner [31] then the global cost of the product by \widehat{S}_λ is of order $O(\frac{1}{dh^2})$.

4.3. Numerical three dimensional experiments. The goal of this section is to demonstrate the very good parallel properties of the mortar element method and the block-diagonal preconditioner described above. Therefore we consider only uniform grids and decompositions into equal subdomains. The equation

$$-\Delta p + \epsilon p = f$$

with $\epsilon = 10^{-4}$ is solved in a parallelepiped. All the computations have been performed on the Cray T3E computer with up to 64 processors used. As a stopping criterion for the iterative method, we want to reduce the preconditioned residual by a factor η while the number of Chebyshev iterations is constant and equal to 8, except when mentioned explicitly. For all computations, the number of multigrid levels is equal to 4, except in section 4.3.1.

Note that even with matched grids at subdomains' interfaces the solution is different from that of the single domain case, because of the mortar element treatment of the interface.

4.3.1. Computing time versus problem size. The unit cube is decomposed into 64 cubic subdomains. In each subdomain the grid is uniform and contains N nodes, N taking the value 25^3 , 33^3 , 41^3 . The total number of nodes varies from 1 000 000 to 4 410 944. The table 1 displays the dependence of the elapsed CPU time and of the number of iterations on N . The desired accuracy is 10^{-7} and the number of Chebyshev iterations is 8 or 16. For these tests, the number of multigrid levels equals 3, and the number of processors is fixed at 64.

TABLE 1. Number of iterations and elapsed CPU time *vs* the number of unknowns in the subdomains and the number of Chebyshev iterations, with 64 processors.

N		25^3	33^3	41^3
8 Cheb. it.	#iter	82	82	82
	T_{cpu}	39	81	141
16 Cheb. it.	#iter	66	68	68
	T_{cpu}	48	97	165

TABLE 2. CPU time (speed-up) *vs* the number of processors and stopping criterion.

Number of processors	16 4 sd./pr.	32 2 sd./pr.	64 1 sd./pr.	Number of iterations	$\ Bu^{k_n}\ $
$\eta = 10^{-5}$	89(1)	45(1.97)	23(3.86)	24	1.5e-5
$\eta = 10^{-6}$	247(1)	124(1.99)	64(3.85)	65	1.1e-6
$\eta = 10^{-7}$	351(1)	177(1.98)	91(3.85)	92	6.5e-8

The CPU time varies slightly sub linearly with the number of unknowns. This can be explained by cache effects when the size of the problem is increased.

4.3.2. *Computing time versus stopping criterion and number of processors.* In Table 2 the elapsed CPU time (in seconds) versus the stopping criterion ε and the number of processors is shown.

We wish to estimate the speed-up of the method, i.e. the dependence of the elapsed CPU time with the number of processors, the global mesh size of the problem and the number of subdomains being fixed. The total number of grid nodes is equal to $129 \times 129 \times 129 = 2\,146\,689$, and the number of subdomains is $4 \times 4 \times 4 = 64$. The subdomains are grouped into 16, 32 or 64 clusters ($N_c = N_c^x \times N_c^y \times N_c^z$), so that $16 = 4 \times 2 \times 2$ (4 subdomains per cluster), $32 = 4 \times 4 \times 2$ (2 subdomains per cluster), $64 = 4 \times 4 \times 4$ (1 subdomains per cluster). In the Table 2 the elapsed CPU time (in seconds) versus the stopping criterion η and the number of processors is shown. In the last column we give the Euclidean norm of Bu^{k_n} , which is nothing but the jump of the computed solution on the interfaces.

The actual speed-up, given in parentheses, is very close to the ideal, demonstrating thus very good parallel properties of the method. The speed-up is estimated with respect to the 16-processors case.

4.3.3. *Scalability.* The next series of results (Table 3) prove the excellent scalability of the algorithm. Now each subdomain is a cube with the edge length 0.5 and has a grid composed of $33 \times 33 \times 33$ nodes. The number of processors used for computation increases linearly with the number of subdomains, so that there is always one subdomain per processor. As we can see from the Table 3, the convergence rate is almost independent on the number of subdomains, although the boundary value problem is not the same, provided the grid in each subdomain does not change, and

TABLE 3. Number of iterations and CPU time *vs* the number of processors and stopping criterion for cubic subdomains.

N_{sd}		16	32	64
$\eta = 10^{-5}$	#iter	27	29	29
	T_{cpu}	23	27	28
$\eta = 10^{-6}$	#iter	43	45	46
	T_{cpu}	37	42	45
$\eta = 10^{-7}$	#iter	67	69	70
	T_{cpu}	58	63	69

TABLE 4. Effect of nonmatching grids.

	case 1	case 2
#iter	82	90
T_{cpu}	81	87

the CPU time increases less than 19%, when the number of processors goes from 16 to 64. This increase of CPU time can be explained by the fact that the average number of mortar sides per subdomain increases with the number of subdomains.

4.3.4. *Nonmatching uniform grids.* In this series of tests, we focus on the effect of nonmatching grids on the performances of the solver. The unit cube is divided into $4 \times 4 \times 4$ subdomains. We compare two cases: in the first case, all the subdomains have a grid with $33 \times 33 \times 33$ nodes, so the grids are matched at the interfaces. In the second case, only the subdomains located in the half space $x_3 > 0.5$ have $33 \times 33 \times 33$ nodes while the other subdomains have $25 \times 25 \times 25$ nodes. For these tests, the number of multigrid levels equals 3, and the number of Chebyshev iterations is 8.

The performances of the solver are slightly affected by the presence of non-matching grids. In this example, the load balancing is very bad, so the CPU time is governed by the processors taking care of the finest grids.

5. Numerical results

5.1. 2D results. The 2D results presented here have been obtained with a characteristic Galerkin scheme, and a mortar element method. Here we have used the stream function-vorticity formulation, so the scheme is different from what has been described above, since the velocity is exactly divergence free. The linearized Stokes problem at each time step is computed by using a variant of the Glowinski-Pironneau algorithm see [17, 2].

Here we present simulations for the flow around a cylinder at Reynolds number 9500. This is a quite unsteady flow, which displays many structures. In Figures 1 and 2, we plot the vorticity at several times steps.

A part of the domain partition is plotted in the figure. The finite element used is an over parametric Q_1 element. The grids in the subdomain are structured and

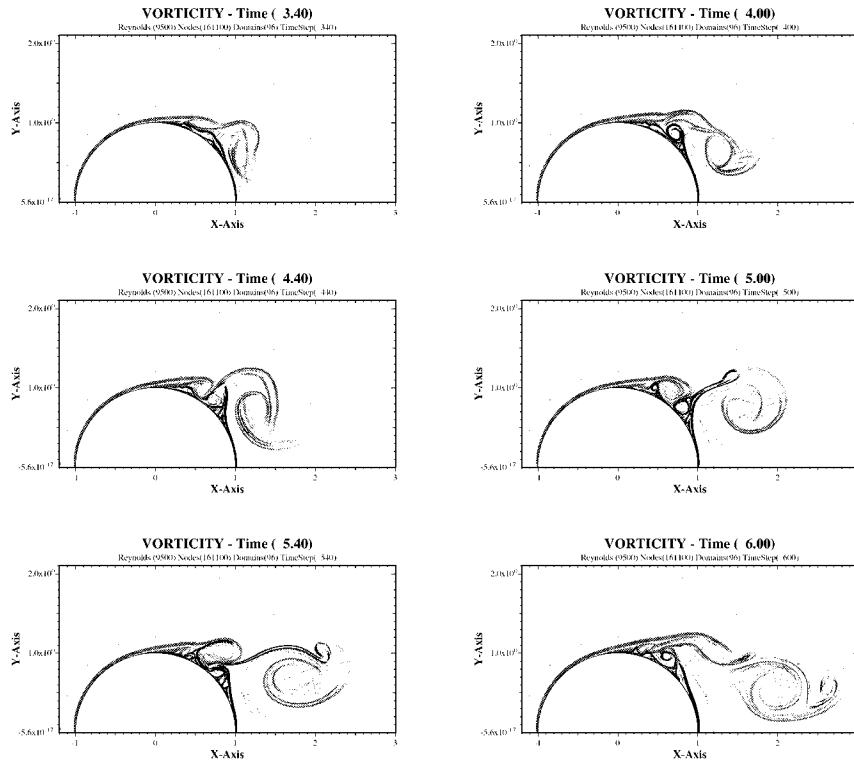


FIGURE 1. Flow around a cylinder at Reynolds number 9500 between $t=3.4s$ and $t=6s$: zoom

refined with a geometrical progression near the walls. The global number of nodes is ~ 160000 and the time step is 0.01. Although it is not seen, the meshes are not matched at the interfaces between the subdomains. However, the grids are fine enough so that there are no visible jumps.

The code is implemented with the parallel library PVM, and is not especially optimized. Such a computations takes two nights on a cluster of 8 HP9000 series700 workstations.

In Figure 3, we have plotted the drag coefficient as a function of time. This coefficient is in good agreement with other computations [21].

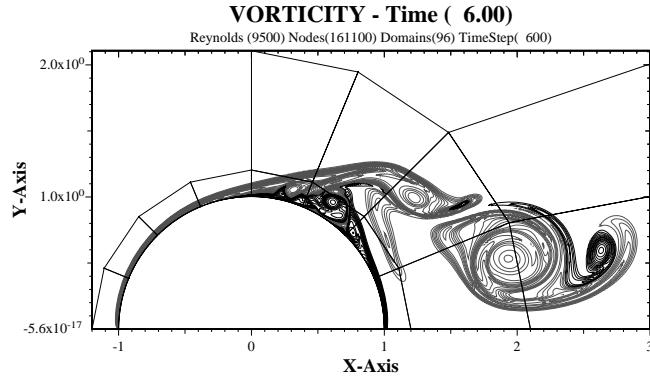


FIGURE 2. Flow around a cylinder at Reynolds number 9500 at time $t=6s$: zoom

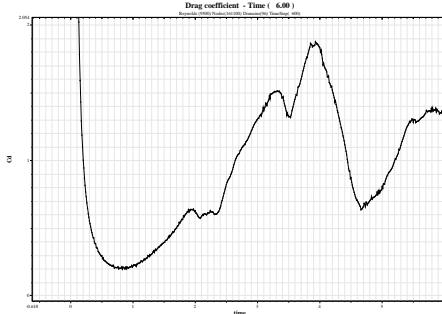


FIGURE 3. Flow around a cylinder at Reynolds number 9500: drag

5.2. 3D results.

5.2.1. *A lid driven cavity at Reynolds number 5000.* This test is concerned with a cubic lid driven cavity at Reynolds number 5000. The flow is driven at the upper wall, where its velocity is 1. On the other walls, a no-slip condition is imposed. The number of grid nodes is 2 100 000. The time step is 0.01. In Figure 4(left), we display the contour lines of the norm of the velocity in the cross-section $y = 0.5$. One can see two principal recirculations in the bottom of the cavity near the corners. The right part of the figure is a crossection in the cross-wind direction: $x = 0.766$. Here we see pairs of vortices in the bottom of the cavity: this instability is called the Taylor-Görtler instability. Note that, at this Reynolds number, this instability is highly unstationary.

The 3D code is written in C++ and parallelized with MPI. Therefore it has been possible to run it on several machines, e.g. PC with Linux, quadriprocessor HP9000, biprocessor Silicon Graphics, IBM SP2, Cray T3E. The present test has

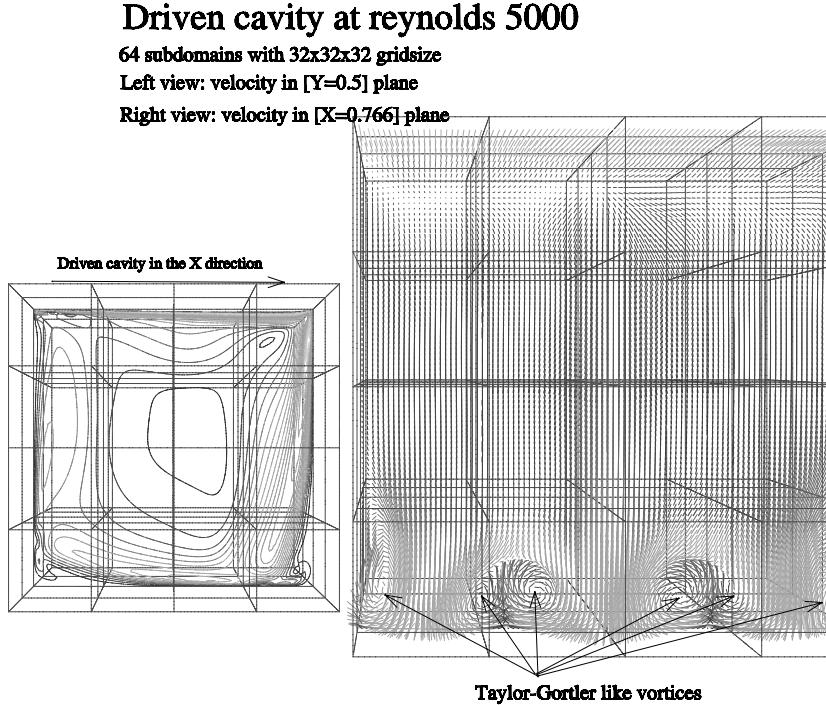


FIGURE 4. Flow in a driven cavity at Reynolds 5000: the Taylor-Görtler instability

been run on the Cray T3E at Idris(Orsay). One particularity of our code is that a single processor can handle several subdomains,

- for portability reasons.
- for load balancing: a discrete optimization algorithm is used to share the load and thus the subdomains between the processors in an optimal way.

There are 64 subdomains and 64 processors are used. Here the grids are matched, but the discretization is still non conforming, because no continuity is imposed at crosspoints and edges. For a single time step, (4 elliptic problems, the convection step, the computation of the gradient of the pressure, and the divergence of the velocity) it takes 55 s.

5.2.2. *Flow behind a cylinder at Reynolds number 20.* This test is concerned with the flow behind a cylinder at Reynolds number 20. There are 144 subdomains, (2 layers of 72 subdomains). We have displayed both the domain partition and the norm of the velocity on a crossection which lies at the interface between the two layers of subdomains: here, the continuity conditions are quite relaxed. However, the flow seems well computed even at the interface. To conclude, the computation seems correct though no quantitative tests have been done yet.

Cylinder at reynolds 20 with 144 subdomains

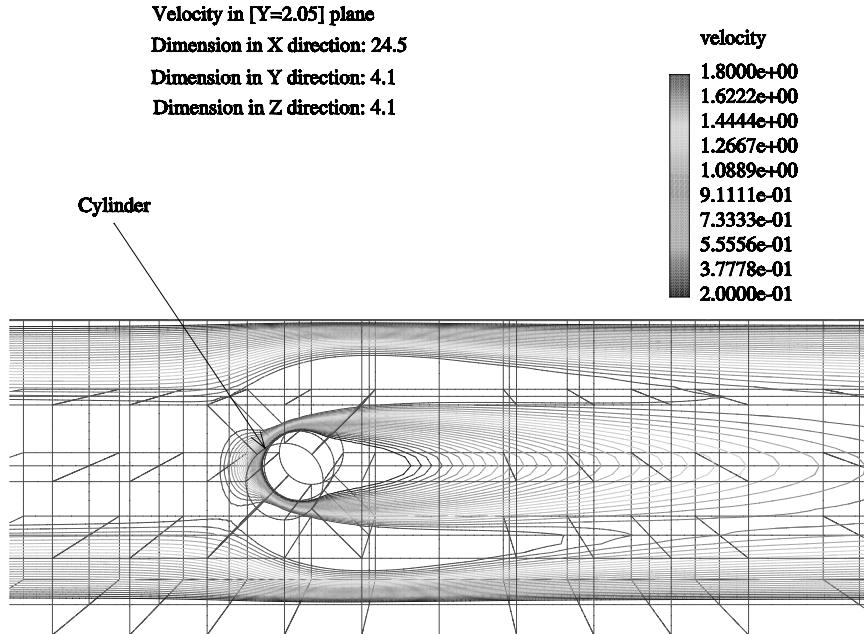


FIGURE 5. Flow behind a cylinder at Reynolds number 20

6. Conclusion

Discretization methods allowing nonmatching grids are useful for many reasons and in particular because they permit the use of locally structured grids. In this paper, we have presented two aspects of the use of nonmatching grids for computational fluid dynamics:

- We have shown how to modify the mortar method to deal with convection-diffusion problems
- We have presented a code in progress for the three dimensional incompressible Navier-Stokes equations.

The code is based on a characteristics-projection scheme, allowing us to use a fast solver developed in [22]. The tests presented above assess the efficiency of the scheme. We have presented also numerical results for high Reynolds flows in 2D and preliminary results in 3D. The results are correct despite the lack of strong continuity but quantitative tests remain to be done. These tests are not complete. In particular, we should investigate if mortar methods are still robust at high Reynolds number with rather coarse meshes, or if additional continuity constraints (continuity at edges or at crosspoints) should be added. We also plan to test the mortar method for compressible fluids.

References

1. G. Abdoulaev, Y. Achdou, Y. Kuznetsov, and C. Prud'homme, *On the parallel implementation of the mortar element method*, (1997), To appear.
2. Y. Achdou and O. Pironneau, *A fast solver for Navier-Stokes equations in the laminar regime using mortar finite element and boundary element method*, SIAM J. Numer. Anal. **32** (1995), 985–1016.
3. Yves Achdou, *The mortar method for convection diffusion problems*, C.R. Acad. Sci. Paris, serie I **321** (1995), 117–123.
4. Yves Achdou and Jean-Luc Guermond, *Convergence analysis of a finite element projection/Lagrange-Galerkin method for the incompressible Navier-Stokes equations*, Tech. Report 96-19, LIMSI, LIMSI CNRS - BP 133 F-91403 ORSAY Cedex (France), December 1996.
5. Yves Achdou, Yvon Maday, and Olof B. Widlund, *Méthode itérative de sous-structuration pour les éléments avec joints*, C.R. Acad. Sci. Paris (1996), no. 322, 185–190.
6. G. Anagnosou, Y. Maday, C. Mavriplis, and A. Patera, *the mortar element method: generalization and implementation*, Proceedings of the Third International Conference on Domain Decomposition Methods for PDE (T. Chan et al, ed.), SIAM, 1990.
7. I. Babuška, *The finite element method with Lagragian multipliers*, Numer. Math. **20** (1973), 179–192.
8. Faker Ben Belgacem, *The mortar element method with Lagrange multipliers*, Université Paul Sabatier, Toulouse, France, 1994.
9. C. Bernardi and Y. Maday, *Raffinement de maillage en éléments finis par la méthode des joints*, C.R. Acad. Sciences, Paris, t 320, serie I, **320** (1995), 373–377.
10. Christine Bernardi, Yvon Maday, and Anthony T. Patera, *A new non conforming approach to domain decomposition: The mortar element method*, Collège de France Seminar (Haim Brezis and Jacques-Louis Lions, eds.), Pitman, 1994, This paper appeared as a technical report about five years earlier.
11. James H. Bramble, Joseph E. Pasciak, and Jinchao Xu, *Parallel multilevel preconditioners*, Math. Comp. **55** (1990), 1–22.
12. F. Brezzi, *On the existence, uniqueness and approximation of saddle point problems arising from Lagrangian multipliers*, R.A.I.R.O. Anal. Numér. **8** (1974), 129–151.
13. J. Cahouet and J.P. Chabard, *Some fast 3-D finite element solvers for generalized Stokes problems*, Int. J. Num. Meth. in Fluids **8** (1988), 269–295.
14. Mario A. Casarin and Olof B. Widlund, *A hierarchical preconditioner for the mortar finite element method*, ETNA **4** (1996), 75–88.
15. A. J. Chorin, *On the convergence of discrete approximations to the Navier–Stokes equations*, Math. Comp. **23** (1969), 341–353.
16. Maksymilian Dryja, *A substructuring preconditioner for the mortar method in 3d*, these proceedings, 1998.
17. R. Glowinski and O. Pironneau, *Numerical method for the first biharmonic equation and for the Stokes problem*, SIAM review **21** (1974), 167–212.
18. M. Griebel, *Multilevel algorithms considered as iterative methods on semidefinite systems*, SIAM J. Sci. Comput. **15** (1994), 604–620.
19. J.L. Guermond, *Some implementations of projection methods for Navier–Stokes equations*, Modél. Math. Anal. Numér. (M²AN) **30** (1996), 637–667.
20. C. Johnson, U. Navert, and J. Pitkaranta, *Finite element method for linear hyperbolic problems*, Comp. Meth. in Appl. Eng. **45** (1984), 285–312.
21. P. Koumoutsakos and A. Leonard, *High resolution simulations of the flow around an impulsively started cylinder using vertex methods*, J. Fluid. Mech. **296** (1995), 1–38.
22. Yuri A. Kuznetsov, *Efficient iterative solvers for elliptic finite element problems on non-matching grids*, Russ. J. Numer. Anal. Math. Modelling **10** (1995), 187–211.
23. Patrick Le Tallec, *Neumann-Neumann domain decomposition algorithms for solving 2D elliptic problems with nonmatching grids*, East-West J. Numer. Math. **1** (1993), no. 2, 129–146.
24. G.I. Marchuk and Y.A. Kuznetsov, *Méthodes itératives et fonctionnelles quadratiques*, Méthodes Mathématiques de L'Informatique : Sur les Méthodes Numériques en Sciences, Physiques et Economiques (G.I. Marchuk J.L. Lions, ed.), vol. 4, Dunod, Paris, 1974.
25. O. Pironneau, *On the transport diffusion algorithm and its application to the Navier–Stokes equations*, Numer Math **38** (1982), 309–332.

26. R. Rannacher, *On Chorin's projection method for the incompressible Navier–Stokes equations*, Lectures Notes in Mathematics, vol. 1530, Springer, Berlin, 1992, pp. 167–183.
27. E. Suli, *Convergence and nonlinear stability of the Lagrange–Galerkin method*, Numer Math **53** (1988), 459–483.
28. P. Le Tallec, T. Sassi, and M. Vidrascu, *Domain decomposition method with nonmatching grids*, Proceedings of DDM 9, AMS, 1994, pp. 61–74.
29. R. Temam, *Une méthode d'approximation de la solution des équations de Navier–Stokes*, Bull. Soc. Math. France **98** (1968), 115–152.
30. Stefan Turek, *Multilevel pressure-Shur complement techniques for the numerical solution of the incompressible Navier–Stokes equations*, Habilitation thesis, Institut fur Angewandte Mathematik, Universitat Heidelberg Im Neuenheimer Feld 294 69210 Heidelberg Germany, 1996, email:tur@gaia.iwr.uni-heidelberg.de.
31. X.Zhang, *Multilevel Schwarz methods*, Numer Math **63** (1992), 521–539.

INSA RENNES, 20 AV DES BUTTES DE COESMES, 35043 RENNES, FRANCE
E-mail address: yves.achdou@insa-rennes.fr

INSTITUTE OF NUMERICAL MATHEMATICS, RUSSIAN ACADEMY OF SCIENCES, UL. GUBKINA 8,
 GSP-1, MOSCOW, RUSSIA

Current address: CRS4, Via N. Sauro 10, Cagliari 09123, Italy
E-mail address: gassan@crs4.it

UNIVERSITÉ PARIS 6, ANALYSE NUMÉRIQUE, 75252 PARIS CEDEX 05, FRANCE

UNIVERSITY OF HOUSTON, DEPARTMENT OF MATHEMATICS, HOUSTON, TEXAS AND INSTITUTE
 OF NUMERICAL MATHEMATICS, RUSSIAN ACADEMY OF SCIENCES, MOSCOW 117334, RUSSIA
E-mail address: kuz@math.uh.edu

UNIVERSITÉ PARIS 6, ANALYSE NUMÉRIQUE, 75252 PARIS CEDEX 05, FRANCE
E-mail address: pironneau@ann.jussieu.fr

LABORATOIRE ASCI, BATIMENT 506, 91403 ORSAY, FRANCE
E-mail address: prudhomm@asci.fr

A Parallel Non-Overlapping Domain-Decomposition Algorithm for Compressible Fluid Flow Problems on Triangulated Domains

Timothy J. Barth, Tony F. Chan, and Wei-Pai Tang

1. Introduction

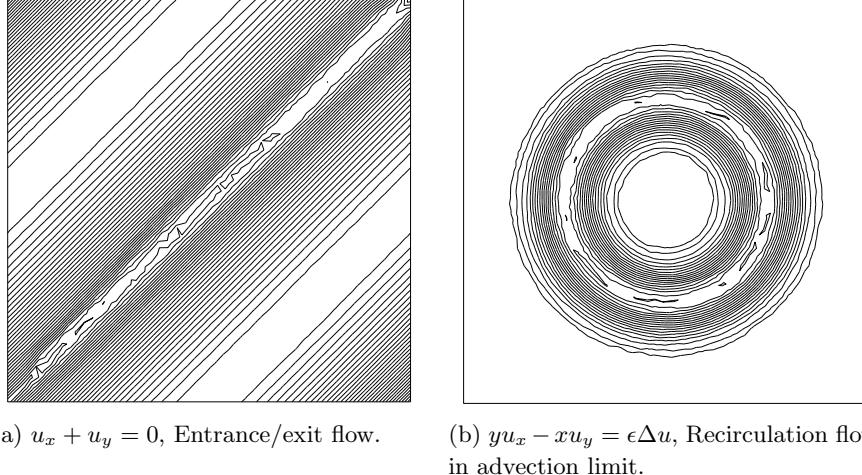
This paper considers an algebraic preconditioning algorithm for hyperbolic-elliptic fluid flow problems. The algorithm is based on a parallel non-overlapping Schur complement domain-decomposition technique for triangulated domains. In the Schur complement technique, the triangulation is first partitioned into a number of non-overlapping subdomains and interfaces. This suggests a reordering of triangulation vertices which separates subdomain and interface solution unknowns. The reordering induces a natural 2×2 block partitioning of the discretization matrix. Exact LU factorization of this block system yields a Schur complement matrix which couples subdomains and the interface together. The remaining sections of this paper present a family of approximate techniques for both constructing and applying the Schur complement as a domain-decomposition preconditioner. The approximate Schur complement serves as an algebraic coarse space operator, thus avoiding the known difficulties associated with the direct formation of a coarse space discretization. In developing Schur complement approximations, particular attention has been given to improving sequential and parallel efficiency of implementations without significantly degrading the quality of the preconditioner. A computer code based on these developments has been tested on the IBM SP2 using MPI message passing protocol. A number of 2-D calculations are presented for both scalar advection-diffusion equations as well as the Euler equations governing compressible fluid flow to demonstrate performance of the preconditioning algorithm.

The efficient numerical simulation of compressible fluid flow about complex geometries continues to be a challenging problem in large scale computing. Many

1991 *Mathematics Subject Classification*. Primary 65Y05; Secondary 65Y10, 76R99, 76N10.

The second author was partially supported by the National Science Foundation grant ASC-9720257, by NASA under contract NAS 2-96027 between NASA and the Universities Space Research Association (USRA).

The third author was partially supported by NASA under contract NAS 2-96027 between NASA and the Universities Space Research Association (USRA), by the Natural Sciences and Engineering Research Council of Canada and by the Information Technology Research Centre which is funded by the Province of Ontario.



(a) $u_x + u_y = 0$, Entrance/exit flow. (b) $yu_x - xu_y = \epsilon \Delta u$, Recirculation flow in advection limit.

FIGURE 1. Two model advection flows.

computational problems of interest in combustion, turbulence, aerodynamic performance analysis and optimization will require orders of magnitude increases in mesh resolution and solution unknowns to adequately resolve relevant fluid flow features. In solving these large problems, algorithmic scalability¹ becomes fundamentally important. To understand algorithmic scalability, we think of the partial differential equation discretization process as producing linear or linearized systems of equations of the form

$$(1) \quad Ax - b = 0$$

where A is some large (usually sparse) matrix, b is a given right-hand-side vector, and x is the desired solution. For many practical problems, the amount of arithmetic computation required to solve (1) by iterative methods can be estimated in terms of the condition number of the system $\kappa(A)$. If A is symmetric positive definite (SPD) the well-known conjugate gradient method converges at a constant rate which depends on κ . After n iterations of the conjugate gradient method, the error ϵ satisfies

$$(2) \quad \frac{\|\epsilon^n\|_2}{\|\epsilon^0\|_2} \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^n.$$

The situation changes considerably for advection dominated problems. The matrix A ceases to be SPD so that the performance of iterative methods is not always linked to the condition number behavior of A . Moreover, the convergence properties associated with A can depend on nonlocal properties of the PDE. To see this, consider the advection and advection-diffusion problems shown in Fig. 1. The entrance/exit flow shown in Fig. 1(a) transports the solution and any error components along 45° characteristics which eventually exit the domain. This is contrasted with the recirculation flow shown in Fig. 1(b) which has circular characteristics in the advection dominated limit. In this (singular) limit, any radially symmetric error components persist for all time. The behavior of iterative methods for these two problems is

¹the arithmetic complexity of algorithms with increasing number of solution unknowns

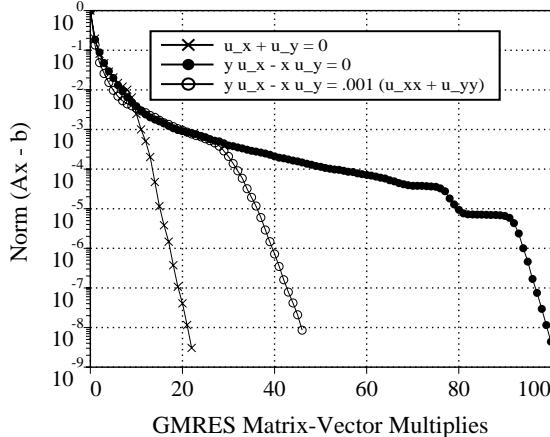


FIGURE 2. Convergence behavior of ILU preconditioned GMRES for entrance/exit and recirculation flow problems using GLS discretization in a triangulated square (1600 dofs).

notably different. Figure 2 graphs the convergence history of ILU-preconditioned GMRES in solving Cuthill-McKee ordered matrix problems for entrance/exit flow and recirculation flow discretized using the Galerkin least-squares (GLS) procedure described in Sec. 2. The entrance/exit flow matrix problem is solved to a 10^{-8} accuracy tolerance in approximately 20 ILU-GMRES iterations. The recirculation flow problem with $\epsilon = 10^{-3}$ requires 45 ILU-GMRES iterations to reach the 10^{-8} tolerance and approximately 100 ILU-GMRES iterations with $\epsilon = 0$. This difference in the number of iterations required for each problem increases dramatically as the mesh is refined. Any theory which addresses scalability and performance of iterative methods for hyperbolic-elliptic problems must address these effects.

2. Stabilized Numerical Discretization of Hyperbolic Systems

Non-overlapping domain-decomposition procedures such as those developed in Sec. 5 strongly motivate the use of compact-stencil spatial discretizations since larger discretization stencils produce larger interface sizes. For this reason, the Petrov-Galerkin approximation due to Hughes, Franca and Mallet [13] has been used in the present study. Consider the prototype conservation law system in m coupled independent variables in the spatial domain $\Omega \subset \mathbf{R}^d$ with boundary surface Γ and exterior normal $\mathbf{n}(x)$

$$(3) \quad \mathbf{u}_t + \mathbf{f}_{,x_i}^i = 0, \quad (x, t) \in \Omega \times [0, \mathbf{R}^+]$$

$$(4) \quad (n_i \mathbf{f}_{\mathbf{u}}^i)^- (\mathbf{u} - \mathbf{g}) = 0, \quad (x, t) \in \Gamma \times [0, \mathbf{R}^+]$$

with implied summation over repeated indices. In this equation, $\mathbf{u} \in \mathbf{R}^m$ denotes the vector of conserved variables and $\mathbf{f}^i \in \mathbf{R}^m$ the inviscid flux vectors. The vector \mathbf{g} can be suitably chosen to impose characteristic data or surface flow tangency using reflection principles. The conservation law system (3) is assumed to possess a generalized entropy pair so that the change of variables $\mathbf{u}(\mathbf{v}) : \mathbf{R}^m \mapsto \mathbf{R}^m$

symmetrizes the system in quasi-linear form

$$(5) \quad \mathbf{u}, \mathbf{v} \mathbf{v}_{,t} + \mathbf{f}_{,\mathbf{v}}^i \mathbf{v}_{,x_i} = 0$$

with \mathbf{u}, \mathbf{v} symmetric positive definite and $\mathbf{f}_{,\mathbf{v}}^i$ symmetric. The computational domain Ω is composed of non-overlapping simplicial elements T_i , $\Omega = \cup T_i$, $T_i \cap T_j = \emptyset$, $i \neq j$. For purposes of the present study, our attention is restricted to steady-state calculations. Time derivatives are retained in the Galerkin integral so that a pseudo-time marching strategy can be used for obtaining steady-state solutions. The Galerkin least-squares method due to Hughes, Franca and Mallet [13] can be defined via the following variational problem with time derivatives omitted from the least-squares bilinear form: Let \mathcal{V}^h denote the finite element space

$$(6) \quad \mathcal{V}^h = \left\{ \mathbf{w}^h \mid \mathbf{w}^h \in \left(C^0(\Omega) \right)^m, \mathbf{w}^h|_T \in \left(\mathcal{P}_k(T) \right)^m \right\}.$$

Find $\mathbf{v}^h \in \mathcal{V}^h$ such that for all $\mathbf{w}^h \in \mathcal{V}^h$

$$(7) \quad B(\mathbf{v}^h, \mathbf{w}^h)_{gal} + B(\mathbf{v}^h, \mathbf{w}^h)_{ls} + B(\mathbf{v}^h, \mathbf{w}^h)_{bc} = 0$$

with

$$\begin{aligned} B(\mathbf{v}, \mathbf{w})_{gal} &= \int_{\Omega} (\mathbf{w}^T \mathbf{u}(\mathbf{v}),_t - \mathbf{w}^T_{,x_i} \mathbf{f}^i(\mathbf{v})) d\Omega \\ B(\mathbf{v}, \mathbf{w})_{ls} &= \sum_{T \in \Omega} \int_T (\mathbf{f}_{,\mathbf{v}}^i \mathbf{w}_{,x_i})^T \boldsymbol{\tau} (\mathbf{f}_{,\mathbf{v}}^i \mathbf{v}_{,x_i}) d\Omega \\ B(\mathbf{v}, \mathbf{w})_{bc} &= \int_{\Gamma} \mathbf{w}^T \mathbf{h}(\mathbf{v}, \mathbf{g}; \mathbf{n}) d\Gamma \end{aligned}$$

where

$$(8) \quad \mathbf{h}(\mathbf{v}_-, \mathbf{v}_+, \mathbf{n}) = \frac{1}{2} (\mathbf{f}(\mathbf{u}(\mathbf{v}_-); \mathbf{n}) + \mathbf{f}(\mathbf{u}(\mathbf{v}_+); \mathbf{n})) - \frac{1}{2} |A(\mathbf{u}(\bar{\mathbf{v}}); \mathbf{n})| (\mathbf{u}(\mathbf{v}_+) - \mathbf{u}(\mathbf{v}_-)).$$

Inserting standard C^0 polynomial spatial approximations and mass-lumping of the remaining time derivative terms, yields coupled ordinary differential equations of the form:

$$(9) \quad D \mathbf{u}_t = \mathcal{R}(\mathbf{u}), \quad \mathcal{R}(\mathbf{u}) : \mathbf{R}^n \rightarrow \mathbf{R}^n$$

or in symmetric variables

$$(10) \quad D \mathbf{u}, \mathbf{v} \mathbf{v}_t = \mathcal{R}(\mathbf{u}(\mathbf{v})),$$

where D represents the (diagonal) lumped mass matrix. In the present study, backward Euler time integration with local time linearization is applied to Eqn. (9) yielding:

$$(11) \quad \left[\frac{1}{\Delta t} D - \left(\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \right)^n \right] (\mathbf{u}^{n+1} - \mathbf{u}^n) = \mathcal{R}(\mathbf{u}^n).$$

The above equation can also be viewed as a modified Newton method for solving the steady state equation $\mathcal{R}(\mathbf{u}) = 0$. For each modified Newton step, a large Jacobian matrix must be solved. In practice Δt is varied as an exponential function $\|\mathcal{R}(\mathbf{u})\|$ so that Newton's method is approached as $\|\mathcal{R}(\mathbf{u})\| \rightarrow 0$. Since each Newton iterate in (11) produces a linear system of the form (1), our attention focuses on this prototype linear form.

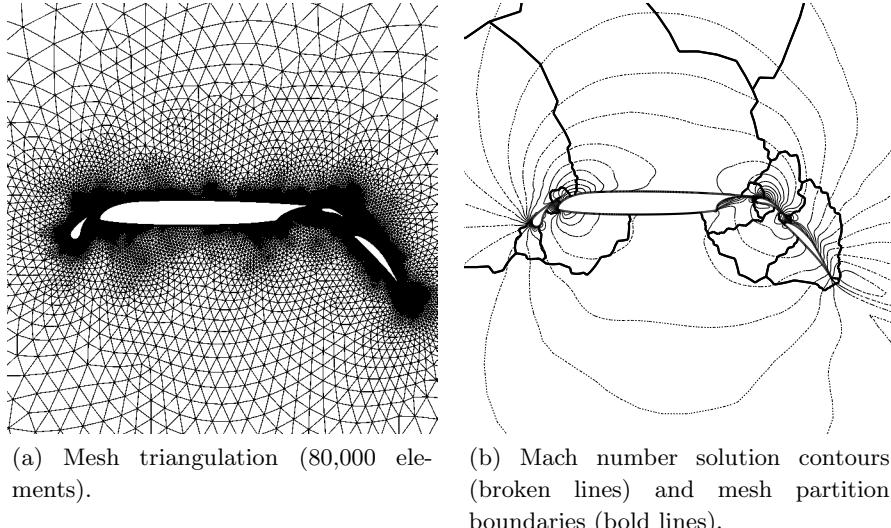


FIGURE 3. Multiple component airfoil geometry with 16 subdomain partitioning and sample solution contours ($M_\infty = .20$, $\alpha = 10^\circ$).

3. Domain Partitioning

In the present study, meshes are partitioned using the multilevel k -way partitioning algorithm METIS developed by Karypis and Kumar [14]. Figure 3(a) shows a typical airfoil geometry and triangulated domain. To construct a non-overlapping partitioning, a dual triangulation graph has been provided to the METIS partitioning software. Figure 3(b) shows partition boundaries and sample solution contours using the spatial discretization technique described in the previous section. By partitioning the dual graph of the triangulation, the number of elements in each subdomain is automatically balanced by the METIS software. Unfortunately, a large percentage of computation in our domain-decomposition algorithm is proportional to the interface size associated with each subdomain. On general meshes containing non-uniform element densities, balancing subdomain sizes does not imply a balance of interface sizes. In fact, results shown in Sec. 6 show increased imbalance of interface sizes as meshes are partitioned into larger numbers of subdomains. This ultimately leads to poor load balancing of the parallel computation. This topic will be revisited in Sec. 6.

4. Matrix Preconditioning

Since the matrix A originating from (11) is assumed to be ill-conditioned, a first step is to consider the prototype linear system in right (or left) preconditioned form

$$(12) \quad (AP^{-1})Px - b = 0.$$

The solution is unchanged but the convergence rate of iterative methods now depends on properties of AP^{-1} . Ideally, one seeks preconditioning matrices P which

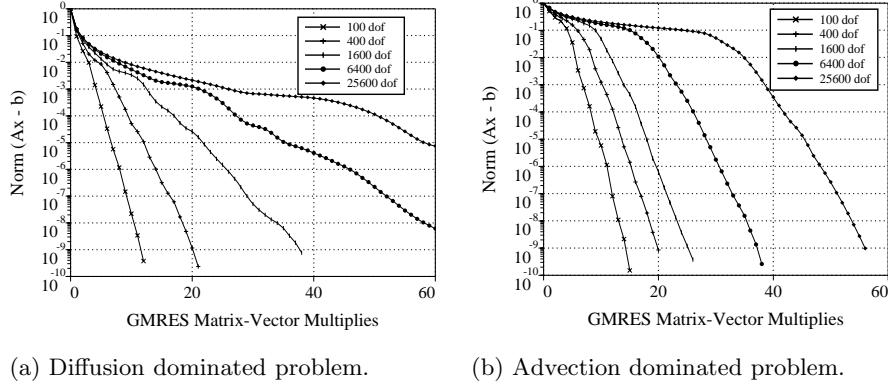


FIGURE 4. Convergence dependence of ILU on the number of mesh points for diffusion and advection dominated problems using SUPG discretization and Cuthill-McKee ordering.

are easily solved and in some sense nearby A , e.g. $\kappa(AP^{-1}) = O(1)$ when A is SPD. Several candidate preconditioning matrices have been considered in this study:

4.1. ILU Factorization. A common preconditioning choice is incomplete lower-upper factorization with arbitrary fill level k , $\text{ILU}[k]$. Early application and analysis of ILU preconditioning is given in Evans [11], Stone [22], and Meijerink and van der Vorst [16]. Although the technique is algebraic and well-suited to sparse matrices, ILU-preconditioned systems are not generally scalable. For example, Dupont *et al.* [10] have shown that $\text{ILU}[0]$ preconditioning does not asymptotically change the $O(h^{-2})$ condition number of the 5-point difference approximation to Laplace's equation. Figure 4 shows the convergence of ILU-preconditioned GMRES for Cuthill-McKee ordered matrix problems obtained from diffusion and advection dominated problems discretized using Galerkin and Galerkin least-squares techniques respectively with linear elements. Both problems show pronounced convergence deterioration as the number of solution unknowns (degrees of freedom) increases. Note that matrix orderings exist for discretized scalar advection equations that are vastly superior to Cuthill-McKee ordering. Unfortunately, these orderings do not generalize naturally to coupled systems of equations which do not have a single characteristic direction. Some ILU matrix ordering experiments are given in [6]. Keep in mind that ILU *does* recluster eigenvalues of the preconditioned matrix so that for small enough problems a noticeable improvement can often be observed when ILU preconditioning is combined with a Krylov projection sequence.

4.2. Multi-level Methods. In the past decade, multi-level approaches such as multigrid has proven to be one of the most effective techniques for solving discretizations of elliptic PDEs [23]. For certain classes of elliptic problems, multigrid attains optimal scalability. For hyperbolic-elliptic problems such as the steady-state Navier-Stokes equations, the success of multigrid is less convincing. For example, Ref. [15] presents numerical results using multigrid to solve compressible Navier-Stokes flow about a multiple-component wing geometry with asymptotic convergence rates approaching .98 (Fig. 12 in Ref. [15]). This is quite far from the usual convergence rates quoted for multigrid on elliptic model problems. This is not too

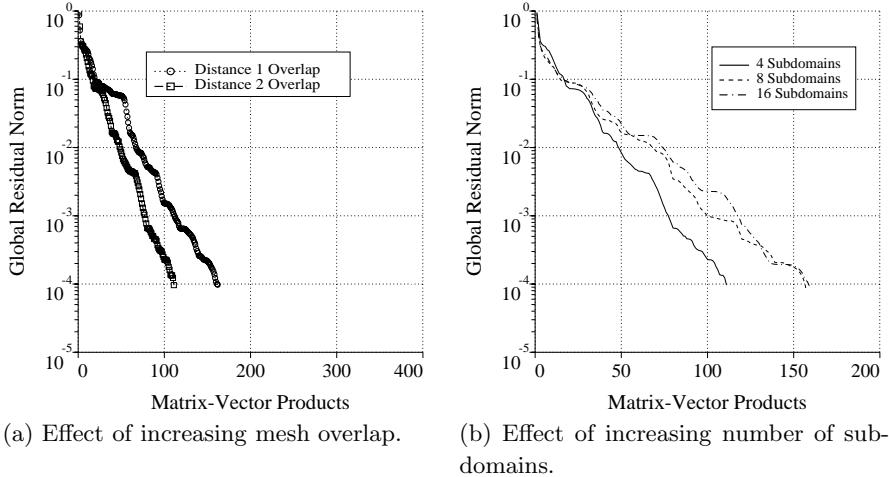


FIGURE 5. Performance of GMRES with additive Schwarz preconditioning.

surprising since multigrid for hyperbolic-elliptic problems is not well-understood. In addition, some multigrid algorithms require operations such as mesh coarsening which are poorly defined for general meshes (especially in 3-D) or place unattainable shape-regularity demands on mesh generation. Other techniques add new meshing constraints to existing software packages which limit the overall applicability of the software.

4.3. Additive Schwarz Methods. The additive Schwarz algorithm [19] is appealing since each subdomain solve can be performed in parallel. Unfortunately the performance of the algorithm deteriorates as the number of subdomains increases. Let H denote the characteristic size of each subdomain, δ the overlap distance, and h the mesh spacing. Dryja and Widlund [8, 9] give the following condition number bound for the method when used as a preconditioner for elliptic discretizations

$$(13) \quad \kappa(AP^{-1}) \leq CH^{-2} \left(1 + (H/\delta)^2\right)$$

where C is a constant independent of H and h . This result describes the deterioration as the number of subdomains increases (and H decreases). With some additional work this deterioration can be removed by the introduction of a global coarse subspace. Under the assumption of “generous overlap” the condition number bound [8, 9, 4] can be improved to

$$(14) \quad \kappa(AP^{-1}) \leq C(1 + (H/\delta)).$$

The addition of a coarse space approximation introduces implementation problems similar to those found in multigrid methods described above. Once again, the theory associated with additive Schwarz methods for hyperbolic PDE systems is not well-developed. Practical applications of the additive Schwarz method for the steady state calculation of hyperbolic PDE systems show similar deterioration of the method when the coarse space is omitted. Figure 5 shows the performance of the additive Schwarz algorithm used as a preconditioner for GMRES. The test matrix

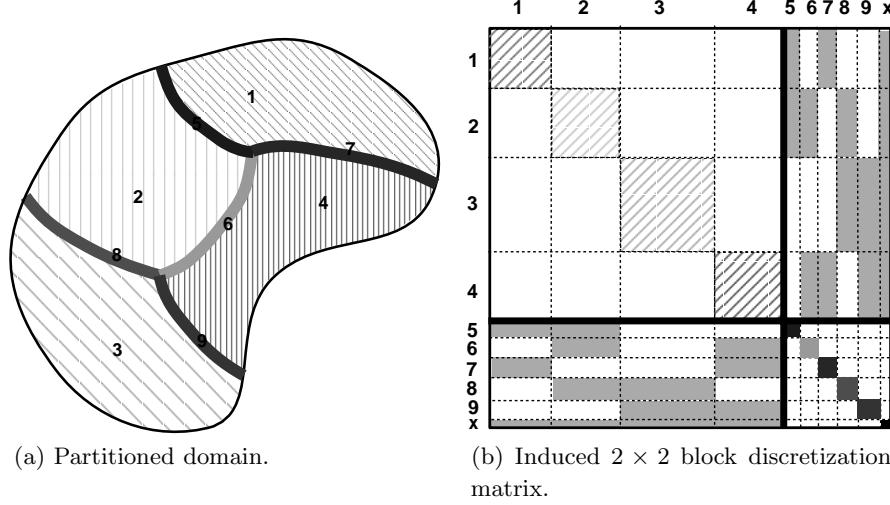


FIGURE 6. Domain decomposition and the corresponding block matrix.

was taken from one step of Newton's method applied to an upwind finite volume discretization of the Euler equations at low Mach number ($M_\infty = .2$), see Barth [1] for further details. These calculations were performed without coarse mesh correction. As expected, the graphs show a degradation in quality with decreasing overlap and increasing number of mesh partitions.

4.4. Schur complement Algorithms. Schur complement preconditioning algorithms are a general family of algebraic techniques in non-overlapping domain-decomposition. These techniques can be interpreted as variants of the well-known substructuring method introduced by Przemieniecki [17] in structural analysis. When recursively applied, the method is related to the nested dissection algorithm. In the present development, we consider an arbitrary domain as illustrated in Fig. 6 that has been further decomposed into subdomains labeled 1 – 4, interfaces labeled 5 – 9, and cross points x . A natural 2×2 partitioning of the system is induced by permuting rows and columns of the discretization matrix so that subdomain unknowns are ordered first, interface unknowns second, and cross points ordered last

$$(15) \quad Ax = \begin{bmatrix} A_{\mathcal{D}\mathcal{D}} & A_{\mathcal{D}\mathcal{I}} \\ A_{\mathcal{I}\mathcal{D}} & A_{\mathcal{I}\mathcal{I}} \end{bmatrix} \begin{pmatrix} x_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix} = \begin{pmatrix} f_{\mathcal{D}} \\ f_{\mathcal{I}} \end{pmatrix}$$

where $x_{\mathcal{D}}, x_{\mathcal{I}}$ denote the subdomain and interface variables respectively. The block LU factorization of A is then given by

$$(16) \quad A = LU = \begin{bmatrix} A_{\mathcal{D}\mathcal{D}} & 0 \\ A_{\mathcal{I}\mathcal{D}} & I \end{bmatrix} \begin{bmatrix} I & A_{\mathcal{D}\mathcal{D}}^{-1}A_{\mathcal{D}\mathcal{I}} \\ 0 & S \end{bmatrix},$$

where

$$(17) \quad S = A_{\mathcal{I}\mathcal{I}} - A_{\mathcal{I}\mathcal{D}}A_{\mathcal{D}\mathcal{D}}^{-1}A_{\mathcal{D}\mathcal{I}}$$

is the Schur complement for the system. Note that $A_{\mathcal{D}\mathcal{D}}$ is block diagonal with each block associated with a subdomain matrix. Subdomains are decoupled from

each other and only coupled to the interface. The subdomain decoupling property is exploited heavily in parallel implementations.

In the next section, we outline a naive parallel implementation of the “exact” factorization. This will serve as the basis for a number of simplifying approximations that will be discussed in later sections.

4.5. “Exact” Factorization. Given the domain partitioning illustrated in Fig. 6, a straightforward (but naive) parallel implementation would assign a processor to each subdomain and a single processor to the Schur complement. Let $\bar{\mathcal{I}}_i$ denote the union of interfaces surrounding \mathcal{D}_i . The entire solution process would then consist of the following steps:

Parallel Preprocessing:

1. Parallel computation of subdomain $A_{\mathcal{D}_i \mathcal{D}_i}$ matrix LU factors.
2. Parallel computation of Schur complement block entries associated with each subdomain \mathcal{D}_i

$$(18) \quad \Delta S_{\bar{\mathcal{I}}_i} = A_{\bar{\mathcal{I}}_i \mathcal{D}_i} A_{\mathcal{D}_i \mathcal{D}_i}^{-1} A_{\mathcal{D}_i \bar{\mathcal{I}}_i}.$$

3. Accumulation of the global Schur complement S matrix

$$(19) \quad S = A_{\mathcal{I} \mathcal{I}} - \sum_{i=1}^{\# \text{subdomains}} \Delta S_{\bar{\mathcal{I}}_i}.$$

Solution:

- Step (1) $u_{\mathcal{D}_i} = A_{\mathcal{D}_i \mathcal{D}_i}^{-1} b_{\mathcal{D}_i}$ (parallel)
- Step (2) $v_{\bar{\mathcal{I}}_i} = A_{\bar{\mathcal{I}}_i \mathcal{D}_i} u_{\mathcal{D}_i}$ (parallel)
- Step (3) $w_{\mathcal{I}} = b_{\mathcal{I}} - \sum_{i=1}^{\# \text{subdomains}} v_{\bar{\mathcal{I}}_i}$ (communication)
- Step (4) $x_{\mathcal{I}} = S^{-1} w_{\mathcal{I}}$ (sequential, communication)
- Step (5) $y_{\mathcal{D}_i} = A_{\mathcal{D}_i \bar{\mathcal{I}}_i} x_{\bar{\mathcal{I}}_i}$ (parallel)
- Step (6) $x_{\mathcal{D}_i} = u_{\mathcal{D}_i} - A_{\mathcal{D}_i \mathcal{D}_i}^{-1} y_{\mathcal{D}_i}$ (parallel)

This algorithm has several deficiencies. Steps 3 and 4 of the solution process are sequential and require communication between the Schur complement and subdomains. More generally, the algorithm is not scalable since the growth in size of the Schur complement with increasing number of subdomains eventually overwhelms the calculation in terms of memory, computation, and communication.

5. Iterative Schur complement Algorithms

A number of approximations have been investigated which simplify the exact factorization algorithm and address the growth in size of the Schur complement. During this investigation, our goal has been to develop algebraic techniques which can be applied to both elliptic and hyperbolic partial differential equations. These approximations include iterative (Krylov projection) subdomain and Schur complement solves, element dropping and other sparsity control strategies, localized subdomain solves in the formation of the Schur complement, and partitioning of the interface and parallel distribution of the Schur complement matrix. Before describing each approximation and technique, we can make several observations:

Observation 1. (Ill-conditioning of Subproblems) For model elliptic problem discretizations, it is known in the two subdomain case that $\kappa(A_{\mathcal{D}_i \mathcal{D}_i}) = O((L/h)^2)$ and $\kappa(S) = O(L/h)$ where L denotes the domain size. From this perspective,

both subproblems are ill-conditioned since the condition number depends on the mesh spacing parameter h . If one considers the scalability experiment, the situation changes in a subtle way. In the scalability experiment, the number of mesh points and the number of subdomains is increased such that the ratio of subdomain size to mesh spacing size H/h is held constant. The subdomain matrices for elliptic problem discretizations now exhibit a $O((H/h)^2)$ condition number so the cost associated with iteratively solving them (with or without preconditioning) is approximately constant as the problem size is increased. Therefore, this portion of the algorithm is scalable. Even so, it may be desirable to precondition the subdomain problems to reduce the overall cost. The Schur complement matrix retains (at best) the $O(L/h)$ condition number and becomes increasingly ill-conditioned as the mesh size is increased. Thus in the scalability experiment, it is ill-conditioning of the Schur complement matrix that must be controlled by adequate preconditioning, see for example Dryja, Smith and Widlund [7].

Observation 2. (Non-stationary Preconditioning) The use of Krylov projection methods to solve the local subdomain and Schur complement subproblems renders the global preconditioner non-stationary. Consequently, Krylov projection methods designed for non-stationary preconditioners should be used for the global problem. For this reason, FGMRES [18], a variant of GMRES designed for non-stationary preconditioning, has been used in the present work.

Observation 3. (Algebraic Coarse Space) The Schur complement serves as an algebraic coarse space operator since the system

$$(20) \quad Sx_{\mathcal{I}} = b_{\mathcal{I}} - A_{\mathcal{ID}}A_{\mathcal{DD}}^{-1}b_{\mathcal{D}}$$

globally couples solution unknowns on the entire interface. The rapid propagation of information to large distances is a crucial component of optimal algorithms.

5.1. ILU-GMRES Subdomain and Schur complement Solves. The first natural approximation is to replace exact inverses of the subdomain and Schur complement subproblems with an iterative Krylov projection method such as GMRES (or stabilized biconjugate gradient).

5.1.1. *Iterative Subdomain Solves.* Recall from the exact factorization algorithm that a subdomain solve is required once in the preprocessing step and twice in the solution step. This suggests replacing these three inverses with m_1 , m_2 , and m_3 steps of GMRES respectively. As mentioned in Observation 1, although the condition number of subdomain problems remains roughly constant in the scalability experiment, it still is beneficial to precondition subdomain problems to improve the overall efficiency of the global preconditioner. By preconditioning subdomain problems, the parameters m_1, m_2, m_3 can be kept small. This will be exploited in later approximations. Since the subdomain matrices are assumed given, it is straightforward to precondition subdomains using ILU[k]. For the GLS spatial discretization, satisfactory performance is achieved using ILU[2].

5.1.2. *Iterative Schur complement Solves.* It is possible to avoid explicitly computing the Schur complement matrix for use in Krylov projection methods by alternatively computing the action of S on a given vector p , i.e.

$$(21) \quad Sp = A_{\mathcal{II}}p - A_{\mathcal{ID}}A_{\mathcal{DD}}^{-1}A_{\mathcal{DI}}p.$$

Unfortunately S is ill-conditioned, thus some form of interface preconditioning is needed. For elliptic problems, the rapid decay of elements away from the diagonal

in the Schur complement matrix [12] permits simple preconditioning techniques. Bramble, Pasciak, and Schatz [3] have shown that even the simple block Jacobi preconditioner yields a substantial improvement in condition number

$$(22) \quad \kappa(SP_S^{-1}) \leq CH^{-2} (1 + \log^2(H/h))$$

for C independent of h and H . For a small number of subdomains, this technique is very effective. To avoid the explicit formation of the diagonal blocks, a number of simplified approximations have been introduced over the last several years, see for examples Bjorstad [2] or Smith [21]. By introducing a further coarse space coupling of cross points to the interface, the condition number is further improved

$$(23) \quad \kappa(SP_S^{-1}) \leq C (1 + \log^2(H/h)).$$

Unfortunately, the Schur complement associated with advection dominated discretizations may not exhibit the rapid element decay found in the elliptic case. This can occur when characteristic trajectories of the advection equation traverse a subdomain from one interface edge to another. Consequently, the Schur complement is not well-preconditioned by elliptic-like preconditioners that use the action of local problems. A more basic strategy has been developed in the present work whereby elements of the Schur complement are *explicitly computed*. Once the elements have been computed, ILU factorization is used to precondition the Schur complement iterative solution. In principle, ILU factorization with a suitable reordering of unknowns can compute the long distance interactions associated with simple advection fields corresponding to entrance/exit-like flows. For general advection fields, it remains a topic of current research to find reordering algorithms suitable for ILU factorization. The situation is further complicated for coupled systems of hyperbolic equations (even in two independent variables) where multiple characteristic directions and/or Cauchy-Riemann systems can be produced. At the present time, Cuthill-McKee ordering has been used on all matrices although improved reordering algorithms are currently under development.

In the present implementation, each subdomain processor computes (in parallel) and stores portions of the Schur complement matrix

$$(24) \quad \Delta S_{\bar{\mathcal{I}}_i} = A_{\bar{\mathcal{I}}_i \mathcal{D}_i} A_{\mathcal{D}_i \mathcal{D}_i}^{-1} A_{\mathcal{D}_i \bar{\mathcal{I}}_i}.$$

To gain improved parallel scalability, the interface edges and cross points are partitioned into a smaller number of generic “subinterfaces”. This subinterface partitioning is accomplished by assigning a supernode to each interface edge separating two subdomains, forming the graph of the Schur complement matrix in terms of these supernodes, and applying the METIS partitioning software to this graph. Let $\bar{\mathcal{I}}_j$ denote the j -th subinterface such that $\mathcal{I} = \cup_j \bar{\mathcal{I}}_j$. Computation of the action of the Schur complement matrix on a vector p needed in Schur complement solves now takes the (highly parallel) form

$$(25) \quad Sp = \sum_{j=1}^{\#\text{subinterfaces}} A_{\bar{\mathcal{I}}_j \bar{\mathcal{I}}_j} p(\bar{\mathcal{I}}_j) - \sum_{i=1}^{\#\text{subdomains}} \Delta S_{\bar{\mathcal{I}}_i} p(\bar{\mathcal{I}}_i).$$

Using this formula it is straightforward to compute the action of S on a vector p to any required accuracy by choosing the subdomain iteration parameter m_i large enough. Figure 7 shows an interface and the immediate neighboring mesh that has been decomposed into 4 smaller subinterface partitions for a 32 subdomain partitioning. By choosing the number of subinterface partitions proportional to

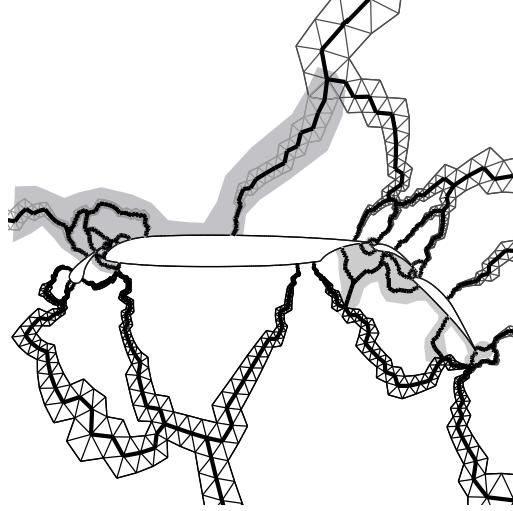


FIGURE 7. Interface (bold lines) decomposed into 4 subinterfaces indicated by alternating shaded regions.

the square root of the number of 2-D subdomains and assigning a processor to each, the number of solution unknowns associated with each subinterface is held approximately constant in the scalability experiment. Note that the use of iterative subdomain solves renders both Eqns. (21) and (25) approximate.

In our investigation, the Schur complement is preconditioned using ILU factorization. This is not a straightforward task for two reasons: (1) portions of the Schur complement are distributed among subdomain processors, (2) the interface itself has been distributed among several subinterface processors. In the next section, a block element dropping strategy is proposed for gathering portions of the Schur complement together on subinterface processors for use in ILU preconditioning the Schur complement solve. Thus, a block Jacobi preconditioner is constructed for the Schur complement which is more powerful than the Bramble, Pasciak, and Schatz (BPS) form (without coarse space correction) since the blocks now correspond to larger subinterfaces rather than the smaller interface edges. Formally, BPS preconditioning without coarse space correction can be obtained for 2D elliptic discretizations by dropping additional terms in our Schur complement matrix approximation and ordering unknowns along interface edges so that the ILU factorization of the tridiagonal-like system for each interface edge becomes exact.

5.1.3. Block Element Dropping. In our implementation, portions of the Schur complement residing on subdomain processors are gathered together on subinterface processors for use in ILU preconditioning of the Schur complement solve. In assembling a Schur complement matrix approximation on each subinterface processor, certain matrix elements are neglected:

1. All elements that couple subinterfaces are ignored. This yields a block Jacobi approximation for subinterfaces.
2. All elements with matrix entry location that exceeds a user specified graph distance from the diagonal as measured on the triangulation graph are ignored. Recall that the Schur complement matrix can be very dense. The

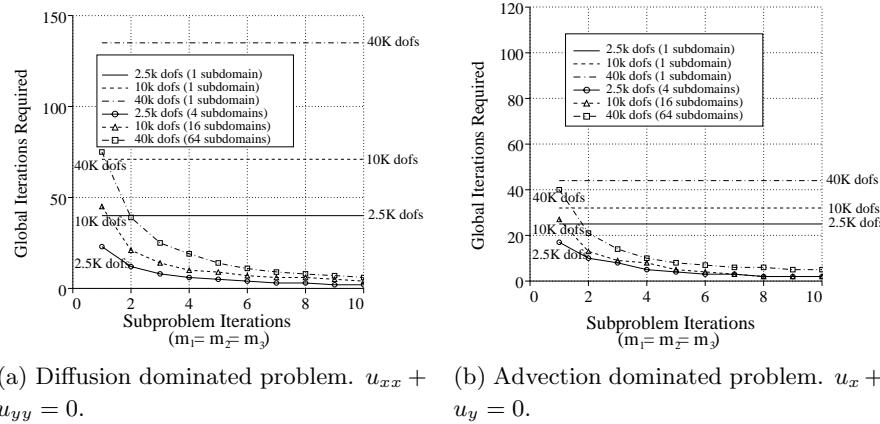
(a) Diffusion dominated problem. $u_{xx} + u_{yy} = 0$.(b) Advection dominated problem. $u_x + u_y = 0$.

FIGURE 8. Effect of the subproblem iteration parameters m_i on the global FGMRES convergence, $m_1 = m_2 = m_3$ for meshes containing 2500, 10000, and 40000 solution unknowns.

graph distance criteria is motivated by the rapid decay of elements away from the matrix diagonal for elliptic problems. In all subsequent calculations, a graph distance threshold of 2 has been chosen for block element dropping.

Figures 8(a) and 8(b) show calculations performed with the present non-overlapping domain-decomposition preconditioner for diffusion and advection problems. These figures graph the number of global FGMRES iterations needed to solve the discretization matrix problem to 10^{-6} accuracy tolerance as a function of the number of subproblem iterations. In this example, all the subproblem iteration parameters have been set equal to each other ($m_1 = m_2 = m_3$). The horizontal lines show poor scalability of single domain ILU-FGMRES on meshes containing 2500, 10000, and 40000 solution unknowns. The remaining curves show the behavior of the Schur complement preconditioned FGMRES on 4, 16, and 64 subdomain meshes. Satisfactory scalability for very small values (5 or 6) of the subproblem iteration parameter m_i is clearly observed.

5.1.4. Wireframe Approximation. A major cost in the explicit construction of the Schur complement is the matrix-matrix product

$$(26) \quad A_{\mathcal{D}_i \mathcal{D}_i}^{-1} A_{\mathcal{D}_i \bar{\mathcal{I}}_i}.$$

Since the subdomain inverse is computed iteratively using ILU-GMRES iteration, forming (26) is equivalent to solving a multiple right-hand sides system with each right-hand side vector corresponding to a column of $A_{\mathcal{D}_i \bar{\mathcal{I}}_i}$. The number of columns of $A_{\mathcal{D}_i \bar{\mathcal{I}}_i}$ is precisely the number of solution unknowns located on the interface surrounding a subdomain. This computational cost can be quite large. Numerical experiments with Krylov projection methods designed for multiple right-hand side systems [20] showed only marginal improvement owing to the fact that the columns are essentially independent. In the following paragraphs, “wireframe” and “super-sparse” approximations are introduced to reduce the cost in forming the Schur complement matrix.

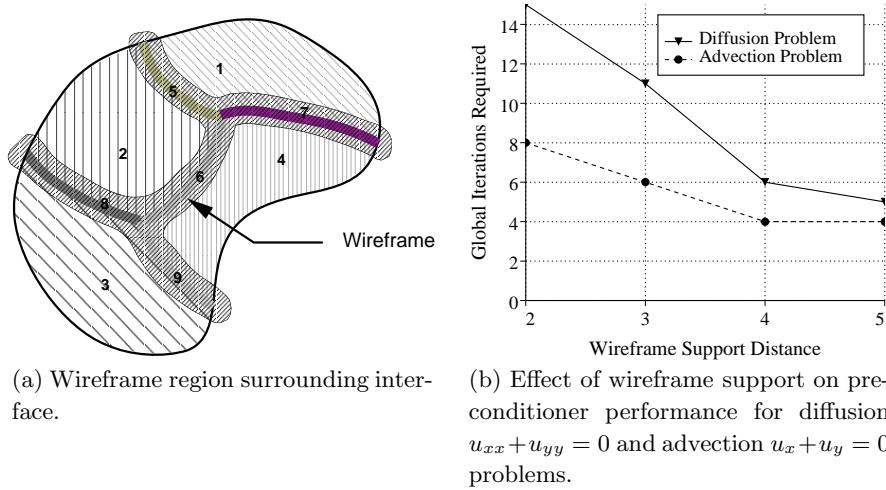


FIGURE 9. Wireframe region surrounding interface and preconditioner performance results for a fixed mesh size (1600 vertices) and 16 subdomain partitioning.

The wireframe approximation idea [5] is motivated from standard elliptic domain-decomposition theory by the rapid decay of elements in S with graph distance from the diagonal. Consider constructing a relatively thin *wireframe* region surrounding the interface as shown in Fig. 9(a). In forming the Eqn. (26) expression, subdomain solves are performed using the much smaller wireframe subdomains. In matrix terms, a principal submatrix of A , corresponding to the variables within the wireframe, is used to compute the (approximate) Schur complement of the interface variables. It is known from domain-decomposition theory that the exact Schur complement of the wireframe region is spectrally equivalent to the Schur complement of the whole domain. This wireframe approximation leads to a substantial savings in the computation of the Schur complement matrix. Note that the full subdomain matrices are used everywhere else in the Schur complement algorithm. The wireframe technique introduces a new adjustable parameter into the preconditioner which represents the width of the wireframe. For simplicity, this width is specified in terms of graph distance on the mesh triangulation. Figure 9(b) demonstrates the performance of this approximation by graphing the total number of preconditioned GMRES iterations required to solve the global matrix problem to a 10^{-6} accuracy tolerance while varying the width of the wireframe. As expected, the quality of the preconditioner improves rapidly with increasing wireframe width with full subdomain-like results obtained using modest wireframe widths. As a consequence of the wireframe construction, the time taken form the Schur complement has dropped by approximately 50%.

5.1.5. Supersparse Matrix-Vector Operations. It is possible to introduce further approximations which improve upon the overall efficiency in forming the Schur complement matrix. One simple idea is to exploit the extreme sparsity in columns of $A_{\mathcal{D}, \overline{\mathcal{I}}_i}$ or equivalently the sparsity in the right-hand sides produced from $A_{\mathcal{D}, \mathcal{D}_i}^{-1} A_{\mathcal{D}_i, \overline{\mathcal{I}}_i}$ needed in the formation of the Schur complement. Observe that m steps of GMRES generates a small sequence of Krylov subspace vectors $[p, A p, A^2 p, \dots, A^m p]$

TABLE 1. Performance of the Schur complement preconditioner with supersparse arithmetic for a 2-D test problem consisting of Euler flow past a multi-element airfoil geometry partitioned into 4 subdomains with 1600 mesh vertices in each subdomain.

Backsolve Fill-Level Distance k	Global GMRES Iterations	Time(k)/Time(∞)
0	26	0.325
1	22	0.313
2	21	0.337
3	20	0.362
4	20	0.392
∞	20	1.000

where p is a right-hand side vector. Consequently for small m , if both A and p are sparse then the sequence of matrix-vector products will be relatively sparse. Standard sparse matrix-vector product subroutines utilize the matrix in sparse storage format and the vector in dense storage format. In the present application, the vectors contain only a few non-zero entries so that standard sparse matrix-vector products waste many arithmetic operations. For this reason, a “supersparse” software library have been developed to take advantage of the sparsity in matrices as well as in vectors by storing both in compressed form. Unfortunately, when GMRES is preconditioned using ILU factorization, the Krylov sequence becomes $[p, AP^{-1}p, (AP^{-1})^2p, \dots, (AP^{-1})^m p]$. Since the inverse of the ILU approximate factors \tilde{L} and \tilde{U} can be dense, the first application of ILU preconditioning produces a dense Krylov vector result. All subsequent Krylov vectors can become dense as well. To prevent this densification of vectors using ILU preconditioning, a fill-level-like strategy has been incorporated into the ILU *backsolve* step. Consider the ILU preconditioning problem, $\tilde{L}\tilde{U}r = b$. This system is conventionally solved by a lower triangular backsolve, $w = \tilde{L}^{-1}b$, followed by a upper triangular backsolve $r = \tilde{U}^{-1}w$. In our supersparse strategy, sparsity is controlled by imposing a non-zero fill pattern for the vectors w and r during lower and upper backsolves. The backsolve fill patterns are most easily specified in terms fill-level distance, i.e. graph distance from existing nonzeros of the right-hand side vector in which new fill in the resultant vector is allowed to occur. This idea is motivated from the element decay phenomena observed for elliptic problems. Table 1 shows the performance benefits of using supersparse computations together with backsolve fill-level specification for a 2-D test problem consisting of Euler flow past a multi-element airfoil geometry partitioned into 4 subdomains with 1600 mesh vertices in each subdomain. Computations were performed on the IBM SP2 parallel computer using MPI message passing protocol. Various values of backsolve fill-level distance were chosen while monitoring the number of global GMRES iterations needed to solve the matrix problem and the time taken to form the Schur complement preconditioner. Results for this problem indicate preconditioning performance comparable to exact ILU backsolves using backsolve fill-level distances of only 2 or 3 with a 60-70% reduction in cost.

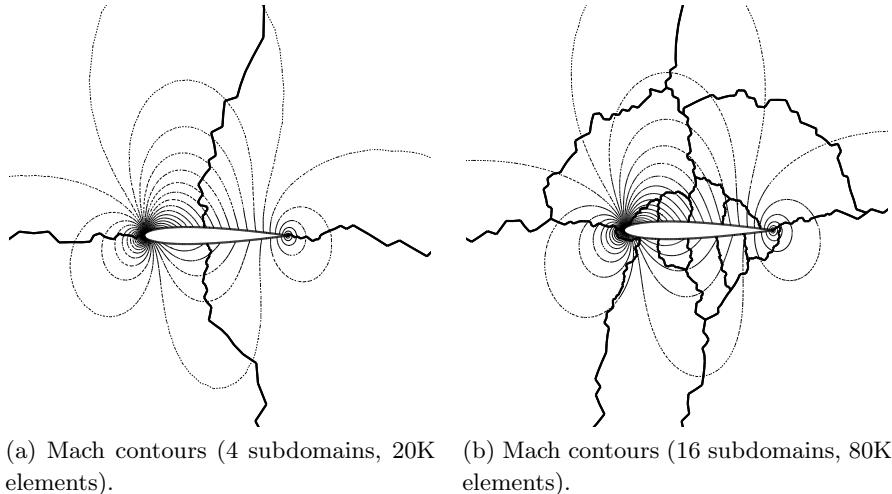


FIGURE 10. Mach number contours and mesh partition boundaries for NACA0012 airfoil geometry.

6. Numerical Results on the IBM SP2

In the remaining paragraphs, we assess the performance of the Schur complement preconditioned FGMRES in solving linear matrix problems associated with an approximate Newton method for the nonlinear discretized compressible Euler equations. All calculations were performed on an IBM SP2 parallel computer using MPI message passing protocol. A scalability experiment was performed on meshes containing 4/1, 16/2, and 64/4 subdomains/subinterfaces with each subdomain containing 5000 mesh elements. Figures 10(a) and 10(b) show mesh partitionings and sample Mach number solution contours for subsonic ($M_\infty = .20$, $\alpha = 2.0^\circ$) flow over the airfoil geometry. The flow field was computed using the stabilized GLS discretization and approximate Newton method described in Sec. 2. Figure 11 graphs the convergence of the approximate Newton method for the 16 subdomain test problem. Each approximate Newton iterate shown in Fig. 11 requires the solution of a linear matrix system which has been solved using the Schur complement preconditioned FGMRES algorithm. Figure 12 graphs the convergence of the FGMRES algorithm for each matrix from the 4 and 16 subdomain test problems. These calculations were performed using ILU[2] and $m_1 = m_2 = m_3 = 5$ iterations on subproblems with supersparse distance equal to 5. The 4 subdomain mesh with 20000 total elements produces matrices that are easily solved in 9-17 global FGMRES iterations. Calculations corresponding to the largest CFL numbers are close approximations to exact Newton iterates. As is typically observed by these methods, the final few Newton iterates are solved more easily than matrices produced during earlier iterates. The most difficult matrix problem required 17 FGMRES iterations and the final Newton iterate required only 12 FGMRES iterations. The 16 subdomain mesh containing 80000 total elements produces matrices that are solved in 12-32 global FGMRES. Due to the nonlinearity in the spatial discretization, several approximate Newton iterates were relatively difficult to solve, requiring over 30 FGMRES iterations. As nonlinear convergence is obtained the matrix problems

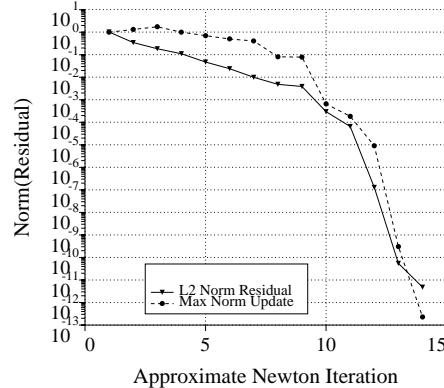


FIGURE 11. Nonlinear convergence behavior of the approximate Newton method for subsonic airfoil flow.

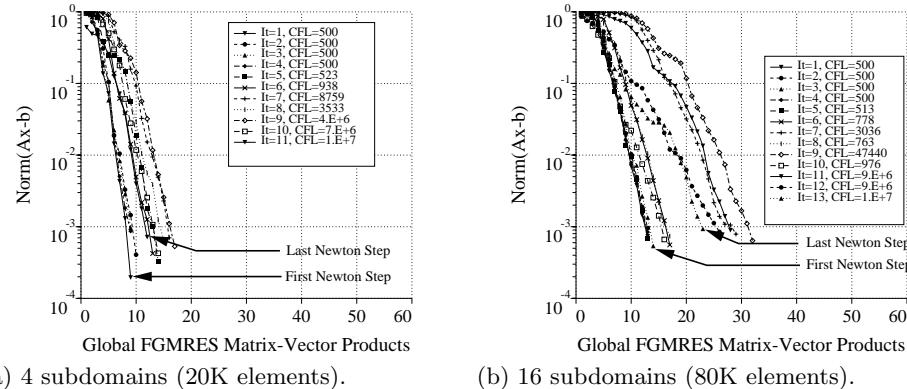


FIGURE 12. FGMRES convergence history for each Newton step.

become less demanding. In this case, the final Newton iterate matrix required 22 FGMRES iterations. This iteration degradation from the 4 subdomain case can be reduced by increasing the subproblem iteration parameters m_1 , m_2 , m_3 but the overall computation time is increased. In the remaining timing graphs, we have sampled timings from 15 FGMRES iterations taken from the final Newton iterate on each mesh. For example, Fig. 13(a) gives a raw timing breakdown for several of the major calculations in the overall solver: calculation of the Schur complement matrix, preconditioning FGMRES with the Schur complement algorithm, matrix element computation and assembly, and FGMRES solve. Results are plotted on each of the meshes containing 4, 16, and 64 subdomains with 5000 elements per subdomain. Since the number of elements in each subdomain is held constant, the time taken to assemble the matrix is also constant. Observe that in our implementation the time to form and apply the Schur complement preconditioner currently dominates the calculation. Although the growth observed in these timings with increasing numbers of subdomains comes from several sources, the dominate effect comes from a very simple source: *the maximum interface size growth associated with subdomains*. This has a devastating impact on the parallel performance since

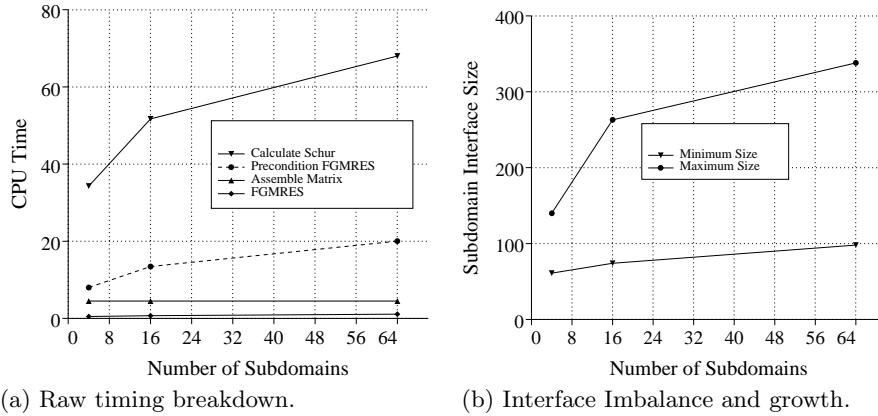


FIGURE 13. Raw IBM SP2 timing breakdown and the effect of increased number of subdomains on smallest and largest interface sizes.

at the Schur complement synchronization point all processors must wait for subdomains working on the largest interfaces to finish. Figure 13(b) plots this growth in maximum interface size as a function of number of subdomains in our scalability experiment. Although the number of elements in each subdomain has been held constant in this experiment, the largest interface associated with any subdomain has more than doubled. This essentially translates into a doubling in time to form the Schur complement matrix. This doubling in time is clearly observed in the raw timing breakdown in Fig. 13(a). At this point in time, we know of no partitioning method that actively addresses controlling the maximum interface size associated with subdomains. We suspect that other non-overlapping methods are sensitive to this effect as well.

7. Concluding Remarks

Experience with our non-overlapping domain-decomposition method with an algebraically generated coarse problem shows that we can successfully trade off some of the robustness of the exact Schur complement method for increased efficiency by making appropriately designed approximations. In particular, the localized wireframe approximation and the supersparse matrix-vector operations together result in reduced cost without significantly degrading the overall convergence rate.

It remains an outstanding problem to partition domains such that the maximum interface size does grow with increased number of subdomains and mesh size. In addition, it may be cost effective to combine this technique with multigrid or multiple-grid techniques to improve the robustness of Newton's method.

References

1. T. J. Barth, *Parallel CFD algorithms on unstructured meshes*, Tech. Report AGARD R-807, Advisory Group for Aeronautical Research and Development, 1995, Special course on parallel computing in CFD.
2. P. Bjorstad and O. B. Widlund, *Solving elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal. **23** (1986), no. 6, 1093–1120.

3. J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), no. 6, 103–134.
4. T. Chan and J. Zou, *Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes*, Tech. Report CAM 93-40, UCLA Department of Mathematics, December 1993.
5. T. F. Chan and T. Mathew, *Domain decomposition algorithms*, Acta Numerica (1994), 61–143.
6. D. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang, *Toward a cost effective ilu preconditioner with high level fill*, BIT **32** (1992), 442–463.
7. M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. **31** (1994), 1662–1694.
8. M. Dryja and O.B. Widlund, *Some domain decomposition algorithms for elliptic problems*, Iterative Methods for Large Linear Systems (L. Hayes and D. Kincaid, eds.), 1989, pp. 273–291.
9. M. Dryja and O.B. Widlund, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, Fifth Conference on Domain Decomposition Methods for Partial Differential Equations (T. F. Chan, D.E. Keyes, G.A. Meurant, J.S. Scroggs, and R.G. Voit, eds.), 1992.
10. T. Dupont, R. Kendall, and H. Rachford, *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal. **5** (1968), 558–573.
11. D. J. Evans, *The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices*, J. Inst. Maths. Applies. **4** (1968), 295–314.
12. G. Golub and D. Mayers, *The use of preconditioning over irregular regions*, Comput. Meth. Appl. Mech. Eng. **6** (1984), 223–234.
13. T. J. R. Hughes, L. P. Franca, and M. Mallet, *A new finite element formulation for CFD: I. symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics*, Comput. Meth. Appl. Mech. Eng. **54** (1986), 223–234.
14. G. Karypis and V. Kumar, *Multilevel k-way partitioning scheme for irregular graphs*, Tech. Report Report 95-064, U. of Minn. Computer Science Department, 1995.
15. D. J. Mavriplis, *A three-dimensional multigrid Reynolds-averaged Navier-Stokes solver for unstructured meshes*, Tech. Report Report 94-29, ICASE, 1994.
16. J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. **34** (1977), 148–162.
17. J. S. Przemieniecki, *Matrix structural analysis of substructures*, Am. Inst. Aero. Astro. J. **1** (1963), 138–147.
18. Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Stat. Comp. **14** (1993), no. 2, 461–469.
19. H. A. Schwarz, *Über einige abbildungsaufgaben*, J. Reine Angew. Math. **70** (1869), 105–120.
20. V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput. **16** (1995), no. 4, 917–933.
21. B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
22. H. Stone, *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Numer. Anal. **5** (1968), 530–558.
23. J. Xu, *An introduction to multilevel methods*, Lecture notes: VIIth EPSRC Numerical Analysis Summer School, 1997.

NASA AMES RESEARCH CENTER, INFORMATION SCIENCES DIRECTORATE, MAIL STOP T27A-1, MOFFETT FIELD, CA 94035

E-mail address: barth@nas.nasa.gov

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA AT LOS ANGELES, LOS ANGELES, CA 90095-1555

E-mail address: chan@math.ucla.edu

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF WATERLOO, WATERLOO, ONTARIO, CANADA N2L 3G1

E-mail address: wptang@bz.uwaterloo.ca

A Non-Overlapping Domain Decomposition Method for the Exterior Helmholtz Problem

Armel de La Bourdonnaye, Charbel Farhat, Antonini Macedo,
Frédéric Magoulès, and François-Xavier Roux

1. Introduction

In this paper, we first show that the domain decomposition methods that are usually efficient for solving elliptic problems typically fail when applied to acoustics problems. Next, we present an alternative domain decomposition algorithm that is better suited for the exterior Helmholtz problem. We describe it in a formalism that can use either one or two Lagrange multiplier fields for solving the corresponding interface problem by a Krylov method. In order to improve convergence and ensure scalability with respect the number of subdomains, we propose two complementary preconditioning techniques. The first preconditioner is based on a spectral analysis of the resulting interface operator and targets the high frequency components of the error. The second preconditioner is based on a coarsening technique, employs plane waves, and addresses the low frequency components of the error. Finally, we show numerically that, using both preconditioners, the convergence rate of the proposed domain decomposition method is quasi independent of the number of elements in the mesh, the number of subdomains, and depends only weakly on the wavenumber, which makes this method uniquely suitable for solving large-scale high frequency exterior acoustics problems.

Acoustic wave propagation problems lead to linear systems that become very large in the high frequency regime. Indeed, for most discretization methods, the mesh size h is typically chosen as one tenth of the wavelength in order to ensure a basic approximation of the physical phenomena. For this reason, many iterative solvers have been and continue to be developed for the Helmholtz problem. In this paper, we consider a domain decomposition based iterative algorithm, because of the success encountered by such methods for the solution of elliptic problems, and because they can be easily implemented on parallel computers.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35J05, 65Y05.

Key words and phrases. Domain decomposition, Lagrange multipliers; exterior Helmholtz problem; Krylov method, preconditioning, coarse grid.

The problem we are interested in solving arises from the discretization of the Helmholtz equation in a bounded domain Ω with an outgoing boundary condition on the outside boundary $\Gamma = \partial\Omega$, and can be written as follows

PROBLEM 1 (The exterior Helmholtz problem). Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$. Find $u \in H^1(\Omega)$ so that

$$(1) \quad -\Delta u - k^2 u = f \text{ in } \Omega$$

$$(2) \quad \frac{\partial u}{\partial n} + \alpha u = f^s \text{ on } \Gamma$$

2. Domain decomposition for exterior acoustics problems

2.1. Classical domain decomposition methods for elliptic problems.

For elliptic problems, non-overlapping domain decomposition methods are usually preferred. In such methods, one splits the initial domain Ω into a finite set of N subdomains Ω_i satisfying

$$(3) \quad \overline{\Omega} = \bigcup_{i=1}^N \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j$$

Let $N_i = \{j, j \neq i, \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\}$, be the set of indices j of the subdomains Ω_j that are neighbors of Ω_i . For such problems, most domain decomposition methods require solving the restriction to each subdomain of the global equation with a set of boundary conditions imposed on the subdomain interfaces. For a suitable choice of boundary conditions and constraints on the subdomain interfaces, each local problem is a well-posed one, and the local solutions u_i in Ω_i are the restrictions to each subdomain of the global solution in Ω .

In the FETI method (cf. C. Farhat and F.-X. Roux [9, 10]), known also as a dual Schur complement method, the following interface conditions are used on $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$

$$(4) \quad \frac{\partial u_i}{\partial n_{ij}} = \lambda_{ij} = -\lambda_{ji} = -\frac{\partial u_j}{\partial n_{ji}} \quad \text{on } \Gamma_{ij} \quad u_i = u_j$$

PROBLEM 2 (The Dual Schur problem). Let $f \in L^2(\Omega)$. Find $u_i \in H^1(\Omega_i)$ satisfying

$$(5) \quad -\Delta u_i = f|_{\Omega_i} \text{ in } \Omega_i$$

$$(6) \quad \frac{\partial u_i}{\partial n_{ij}} = \lambda_{ij} \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

under the constraint

$$(7) \quad u_i - u_j = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

On the other hand, the primal Schur complement method (cf. P. le Tallec [12]) uses a Dirichlet boundary condition on the subdomain interfaces, which ensures the continuity of the solution through these interfaces with the constraint

$$(8) \quad \frac{\partial u_i}{\partial n_{ij}} + \frac{\partial u_j}{\partial n_{ji}} = 0 \text{ sur } \Gamma_{ij} \quad \forall j \in N_i$$

For such a method and in each subdomain, one has

PROBLEM 3 (Primal Schur problem). Let $f \in L^2(\Omega)$. Find $u_i \in H^1(\Omega_i)$ satisfying

$$(9) \quad -\Delta u_i = f|_{\Omega_i} \text{ in } \Omega_i$$

$$(10) \quad u_i = p_{ij} (= p_{ji}) \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

under the given constraint

$$(11) \quad \frac{\partial u_i}{\partial n_{ij}} + \frac{\partial u_j}{\partial n_{ji}} = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

REMARK 4. The local solutions obtained by any of the above methods satisfy the following continuity equations

$$(12) \quad u_i - u_j = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

$$(13) \quad \frac{\partial u_i}{\partial n_{ij}} + \frac{\partial u_j}{\partial n_{ji}} = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

These two equalities ensure that the function u which is equal to u_i in each subdomain Ω_i is the solution of the global problem in $H^1(\Omega)$.

If any of the two domain decomposition methods presented above is used for solving the Helmholtz problem, the associated local problems can become ill-posed when the wavenumber k of the given global problem corresponds to a resonant frequency of the subdomain Laplace operator. It follows that the interface boundary conditions characteristic of domain decomposition methods for strongly elliptic problems cannot be used for the Helmholtz equation (see also [1]).

In [3] and [4], B. Després presents a domain decomposition method for the Helmholtz problem where the local subproblems are well-posed, but where a simple (and rather inefficient) relaxation-like iterative method is employed for solving the resulting interface problem. In this paper, we formulate the interface problem in terms of Lagrange multipliers, and develop a scalable preconditioned Krylov method for solving it.

2.2. A new domain decomposition method for the Helmholtz problem. One way to generate well-posed local problems consists in moving the spectrum of the operator associated to the Helmholtz equation in each subdomain into the complex plane. For example, one can replace the standard Dirichlet or Neumann boundary conditions on the subdomain interfaces by the Robin boundary conditions Ω_i and Ω_j , $\forall j \in N_i$, can be written as

$$(14) \quad \frac{\partial u_i}{\partial n} + iku_i = \lambda_{ij} \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

$$(15) \quad \frac{\partial u_j}{\partial n} + iku_j = \lambda_{ji} \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

where $\lambda_{ij} - \lambda_{ji} = 0$ and n is either n_{ij} or n_{ji} . The constraint on the subdomain interfaces is determined so that local solutions u_i and u_j satisfy the continuity relations (12) and (13). Hence, this constraint can be formulated as

$$(16) \quad \left[\frac{\partial u}{\partial n} - iku \right] = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

where the brackets $[\dots]$ denote the jump of the enclosed quantity through the interface Γ_{ij} between two subdomains. It follows that the problem to be solved in each subdomain is

PROBLEM 5. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$. Find $u_i \in H^1(\Omega_i)$ satisfying

$$(17) \quad -\Delta u_i - k^2 u_i = f|_{\Omega_i} \text{ in } \Omega_i$$

$$(18) \quad \frac{\partial u_i}{\partial n} + iku_i = \lambda_{ij} \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

$$(19) \quad \frac{\partial u_i}{\partial n} + \alpha u_i = f^s \text{ on } \Gamma \cap \partial \Omega_i$$

with the constraint

$$(20) \quad [\frac{\partial u}{\partial n} - iku] = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

Strictly speaking, the above method is correct only for a slice-wise or a checkerboard-like decomposition of Ω . For arbitrary mesh partitions, the above method is not guaranteed to generate well-posed local problems, unless additional precaution is taken [7]. Indeed, the variational formulation of problem (5) is

$$(21) \quad \begin{aligned} \forall v_i \in H^1(\Omega_i), \quad & \int_{\Omega_i} \nabla u_i \nabla v_i - k^2 \int_{\Omega_i} u_i v_i + \alpha \int_{\Gamma \cap \partial \Omega_i} u_i v_i \\ &= \int_{\Omega_i} f_i v_i + \int_{\Gamma \cap \partial \Omega_i} f_i^s v_i + \sum_{j \in N_i} \int_{\Gamma_{ij}} \frac{\partial u_i}{\partial n_{ij}} v_i \end{aligned}$$

Substituting in the above equation the normal derivative with the expression derived from formula (18) leads to the following *Lax-Milgram lemma* bilinear form

$$(22) \quad \int_{\Omega_i} \nabla u_i \nabla v_i - k^2 \int_{\Omega_i} u_i v_i + \alpha \int_{\Gamma \cap \partial \Omega_i} u_i v_i - ik \sum_{j \in N_i} (-1)^{\delta_{n_{ij}}} \int_{\Gamma_{ij}} u_i v_i$$

where $\delta_{n_{ij}}$ is equal to 1 if n is the outgoing normal unit vector to Ω_i , and is equal to 0 otherwise. The H^1 -Ellipticity of the functional is then not satisfied for some partitions of the domain Ω and the problem becomes locally ill-posed. However, as shown in [7] and [5], coloring techniques can be used to extend the domain decomposition method proposed above to arbitrary mesh partitions while ensuring well-posed local problems. Alternatively, one can address general partitions of Ω by relaxing the equality $\lambda_{ij} - \lambda_{ji} = 0$ and introducing independent Lagrange multipliers λ_{ij} and λ_{ji} . In that case, in each subdomain Ω_i , n is chosen as the outgoing unit normal vector, and the global constraint is modified so that in each subdomain the following problem is solved

PROBLEM 6. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$. Find $u_i \in H^1(\Omega_i)$ so that

$$(23) \quad -\Delta u_i - k^2 u_i = f|_{\Omega_i} \text{ in } \Omega_i$$

$$(24) \quad \frac{\partial u_i}{\partial n_{ij}} + iku_i = \lambda_{ij} \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

$$(25) \quad \frac{\partial u_i}{\partial n_i} + \alpha u_i = f^s \text{ on } \Gamma \cap \partial \Omega_i$$

with the double constraint

$$(26) \quad \left[\frac{\partial u}{\partial n_{ij}} + iku \right] = 0 \quad \text{and} \quad \left[\frac{\partial u}{\partial n_{ji}} + iku \right] = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

Note that the above double constraint can be derived from linear combinations of the continuity relations (12) and (13). Indeed, one has

$$(26a) = (13) + ik(12)$$

$$(26b) = (13) - ik(12)$$

Proceeding this way, we note that by inverting (26a) and (26b), the following alternative double constraint is obtained

$$(27) \quad \left[\frac{\partial u}{\partial n_{ij}} - iku \right] = 0 \quad \text{and} \quad \left[\frac{\partial u}{\partial n_{ji}} - iku \right] = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

Nevertheless, we prefer formula (26) because it leads to a linear system of interface equations whose spectrum and conditioning properties are preferable for iterative solution methods (see [8]).

REMARK 7. The domain decomposition method presented here is correctly defined for a strictly positive wavenumber k , but becomes singular when k goes to 0. In the latter case, the double constraint becomes

$$(28) \quad \left[\frac{\partial u}{\partial n_{ij}} \right] = 0 \quad \text{and} \quad \left[\frac{\partial u}{\partial n_{ji}} \right] = 0 \text{ on } \Gamma_{ij} \quad \forall j \in N_i$$

and relation (12) coupling the traces of the local solutions on the interfaces is never satisfied. It follows that the method presented in this section cannot be used for solving the Laplace equation.

NOTATION 8. In the following, we denote by Q_{ij} the operator

$$(29) \quad Q_{ij} : \frac{\partial u_i}{\partial n_{ij}} + iku_i \mapsto \frac{\partial u_i}{\partial n_{ji}} - iku_i$$

It is shown in [2] that Q_{ij} is a unitary operator. Furthermore, its spectrum has an accumulation point at 1. Numerically, the density around this point is inversely proportional to the wavenumber k .

3. Variational formulation of the proposed domain decomposition method

3.1. A Lagrange multiplier formulation. In order to analyze the domain decomposition method presented in this paper, we begin by rewriting the Helmholtz problem (1) as a hybrid problem with two Lagrange multiplier fields on the subdomain interfaces. Following [11] for the Laplace equation, we write the variational formulation of our target problem as follows. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$, find $u \in H^1(\Omega)$ so that

$$(30) \quad \forall v \in H^1(\Omega), \quad \int_{\Omega} \nabla u \nabla v - k^2 \int_{\Omega} uv + \alpha \int_{\Gamma} uv = \int_{\Omega} fv + \int_{\Gamma} f^s v$$

Next, we consider a decomposition of Ω into N subdomains

$$(31) \quad \overline{\Omega} = \bigcup_{i=1}^N \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j$$

and rewrite the variational formulation (30) as follows. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$. Find $u \in H^1(\Omega)$ so that

$$(32) \quad \begin{aligned} \forall v \in H^1(\Omega), \quad & \sum_{i=1}^N \int_{\Omega_i} \nabla(u|_{\Omega_i}) \nabla(v|_{\Omega_i}) - k^2 \int_{\Omega_i} (u|_{\Omega_i})(v|_{\Omega_i}) + \alpha \int_{\Gamma \cap \partial \Omega_i} (u|_{\Omega_i})(v|_{\Omega_i}) \\ & = \sum_{i=1}^N \int_{\Omega_i} (f|_{\Omega_i})(v|_{\Omega_i}) + \int_{\Gamma \cap \partial \Omega_i} (f^s|_{\Omega_i})(v|_{\Omega_i}) \end{aligned}$$

where $(v|_{\Omega_i})$ is the restriction of v to Ω_i . Instead of looking for a function u defined in Ω , it is easier to look for an N -tuple $u^* = (u_1, \dots, u_N)$ belonging to a space V^* spanned by these restrictions.

$$(33) \quad V^* = \{v^* = (v_1, \dots, v_N), \exists v \in H^1(\Omega), \forall i, 1 \leq i \leq N, v_i = v|_{\Omega_i}\}$$

The space V^* can be written in terms of the space X^* which is the product of the spaces X_i , defined by

$$(34) \quad X_i = \{v_i \in H^1(\Omega_i)\}, \quad X^* = \prod_{i=1}^N X_i$$

as follows

$$(35) \quad V^* = \{v^* = (v_1, \dots, v_N) \in X^*, \forall i, 1 \leq i \leq N, \forall j \in N_i, v_i|_{\Gamma_{ij}} = v_j|_{\Gamma_{ij}}\}$$

where $v_i|_{\Gamma_{ij}}$ is the trace on the interface Γ_{ij} of the function v_i . The constraint on the subdomain interfaces can be relaxed by introducing a double Lagrange multiplier $(\lambda_{ij}, \lambda_{ji})$ in the equation as presented in the previous section. This Lagrange multiplier belongs to the space M included in $\prod_{1 \leq i \leq N} \prod_{j \in N_i} H^{-\frac{1}{2}}(\Gamma_{ij})$. The initial problem is thus equivalent to the following constrained problem

PROBLEM 9. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$, find $u \in V^*$ so that :

$$(36) \quad \begin{aligned} \forall v \in V^*, \quad & , \sum_{i=1}^N \int_{\Omega_i} \nabla u_i \nabla v_i - k^2 \int_{\Omega_i} u_i v_i + \alpha \int_{\Gamma \cap \partial \Omega_i} u_i v_i \\ & = \sum_{i=1}^N \int_{\Omega_i} f_i v_i + \int_{\Gamma \cap \partial \Omega_i} f_i^s v_i \end{aligned}$$

and, with the notation introduced above, to the hybrid problem

PROBLEM 10. Let $f \in L^2(\Omega)$ and $f^s \in L^2(\Gamma)$, find $(u^*, \lambda) \in X^* \times M$ so that

$$(37) \quad \begin{aligned} \forall v^* \in X^*, \quad & \sum_{i=1}^N \int_{\Omega_i} \nabla u_i \nabla v_i - k^2 \int_{\Omega_i} u_i v_i + \alpha \int_{\Gamma \cap \partial \Omega_i} u_i v_i \\ & + \sum_{j \in N_i} \int_{\Gamma_{ij}} (-\lambda_{ij} + iku_i|_{\Gamma_{ij}}) v_i|_{\Gamma_{ij}} = \sum_{i=1}^N \int_{\Omega_i} f_i v_i + \int_{\Gamma \cap \partial \Omega_i} f_i^s v_i \\ \forall v \in X^*, \quad & \sum_{i=1}^N \sum_{j \in N_i} \int_{\Gamma_{ij}} (\lambda_{ij} + \lambda_{ji} - 2iku_i|_{\Gamma_{ij}}) v_i|_{\Gamma_{ij}} = 0 \end{aligned}$$

$$(38) \quad \sum_{i=1}^N \sum_{j \in N_i} \int_{\Gamma_{ij}} (\lambda_{ij} + \lambda_{ji} - 2ik u_j|_{\Gamma_{ij}}) v_i|_{\Gamma_{ij}} = 0$$

In the sequel, we discretize problem (10) for conforming meshes using the method of [10].

3.2. Discretization of the governing equations. Discretizing the hybrid formulation derived in the previous section leads to

$$(39) \quad A_i u_i = f_i + \sum_{j \in N_i} B_{ij}^t \lambda_{ij}$$

where the matrix A_i results from the discretization of the bilinear form

$$(40) \quad \int_{\Omega_i} \nabla u_i \nabla v_i - k^2 \int_{\Omega_i} u_i v_i + \alpha \int_{\Gamma \cup \partial \Omega_i} u_i v_i + ik \sum_{j \in N_i} \int_{\Gamma_{ij}} u_i|_{\Gamma_{ij}} v_i|_{\Gamma_{ij}}$$

and B_{ij} represents the operator

$$(41) \quad (B_{ij} u_i, \lambda_{ij}) = (u_i, B_{ij}^t \lambda_{ij}) = \int_{\Gamma_{ij}} \lambda_{ij} u_i|_{\Gamma_{ij}}$$

The question that arises first is that of the choice of the space of discretization for the constraint variables λ_{ij} and λ_{ji} . But, as our problem is not exactly a hybrid one, there is no way to fulfil the *Ladyzhenskaya-Babuska-Brezzi* condition uniformly. As argued in the next remark, this is not a problem in our case, and following the analysis presented in [10], we can choose for the operator B_{ij} the restriction to the interface of the operator R_{ij} . Such an approach corresponds to choosing for the constraint fields the space of the Dirac masses that are centered on the nodes of the subdomain interfaces. Let us recall that this choice ensures the equality of the discrete fields on the interface for the Laplace equation. Hence, if one defines M_{ij} as the mass matrix on the subdomain interfaces

$$(42) \quad (M_{ij} R_{ij} u_i, R_{ji} u_j) = (R_{ji}^t M_{ij} R_{ij} u_i, u_j) = \int_{\Gamma_{ij}} u_i|_{\Gamma_{ij}} u_j|_{\Gamma_{ij}}$$

one can discretize the first constraint of problem (10)

$$(43) \quad \int_{\Gamma_{ij}} (\lambda_{ij} + \lambda_{ji} - 2ik u_i|_{\Gamma_{ij}}) u_j|_{\Gamma_{ij}} = 0$$

as

$$(44) \quad (\lambda_{ij} + \lambda_{ji} - 2ik M_{ij} R_{ij} u_i, R_{ji} u_j) = 0$$

which can be rewritten as

$$(45) \quad R_{ji}^t (\lambda_{ij} + \lambda_{ji} - 2ik M_{ij} R_{ij} u_i) = 0$$

Taking into account that $R_{ij} = [0 \quad I]$, one can deduce

$$(46) \quad \lambda_{ij} + \lambda_{ji} - 2ik M_{ij} R_{ij} u_i = 0$$

Applying the same treatment to the second constraint, one finally obtains

$$(47) \quad \sum_{i=1}^N \sum_{j \in N_i} \lambda_{ij} + \lambda_{ji} - 2ik M_{ij} R_{ij} u_i = 0$$

$$(48) \quad \sum_{i=1}^N \sum_{j \in N_i} \lambda_{ij} + \lambda_{ji} - 2ikM_{ji}R_{ji}u_j = 0$$

REMARK 11. Adding equations (47) and (48), one obtains the following system of equations

$$(49) \quad \sum_{i=1}^N \sum_{j \in N_i} -2ikM_{ij}(R_{ij}u_i - R_{ji}u_j) = 0$$

and thus

$$(50) \quad (R_{ij}u_i - R_{ji}u_j) = 0 \quad \forall i \text{ and } j \in N_i$$

In other words, the discrete fields u_i are continuous across the subdomain interfaces. Hence, one can assemble the local equations and show that the local discrete solutions in each subset Ω_i are indeed the restrictions of the discrete solutions of the global problem since there is no approximation in the way the constraints are satisfied.

Substituting u_i and u_j obtained in equation (39) into equation (47) and equation (48), leads to the following interface problem

$$(51) \quad \sum_{i=1}^N \sum_{j \in N_i} \lambda_{ij} + \lambda_{ji} - 2ikM_{ij}R_{ij}A_i^{-1}R_{ij}^t\lambda_{ij} = 2ikM_{ij}R_{ij}A_i^{-1}f_i$$

$$(52) \quad \sum_{i=1}^N \sum_{j \in N_i} \lambda_{ij} + \lambda_{ji} - 2ikM_{ji}R_{ji}A_j^{-1}R_{ji}^t\lambda_{ji} = 2ikM_{ji}R_{ji}A_j^{-1}f_j$$

Denoting by λ the double Lagrange multiplier $(\lambda_{ij}, \lambda_{ji})$ defined on all the interfaces, the previous system can be written as

$$(53) \quad D\lambda = b$$

where D is a dense, complex, regular, unsymmetric and non hermitian matrix. This matrix is not explicitly known, and assembling it is computationally inefficient. However, given that D is the sum over the subdomains of local matrices, its product by a vector needs only local data (see Section 2.3.2). For these reasons, an iterative method is the most suitable method for solving the linear system $D\lambda = b$.

3.3. Iterative solution of the the hybrid problem.

3.3.1. *The Generalized Conjugated Residuals algorithm.* Among all iterative methods, the *Generalized Conjugated Residuals* algorithm (GCR) is perhaps the most efficient for solving the linear system $D\lambda = b$ where D is a complex matrix. This algorithm minimizes $\|D\lambda - b\|^2$ on the Krylov spaces $K_{p+1} = \{g_0, Dg_0, \dots, D^pg_0\}$ with growing dimension p .

Knowing at iteration p the approximate solution λ^p , the residual $g^p = D\lambda^p - b$, the normalized descent direction vectors $\{w^0, \dots, w^p\}$ and their product by matrix D , $\{Dw^0, \dots, Dw^p\}$, the iteration $p + 1$ d of the GCR algorithm goes as follows

- Compute the optimal descent coefficient

$$(54) \quad \rho^p = -(g^p, \overline{Dw^p})$$

- Update the solution and the residual

$$(55) \quad \lambda^{p+1} = \lambda^p + \rho^p w^p$$

$$(56) \quad g^{p+1} = g^p + \rho^p D w^p$$

- Compute the product of Dw^p by matrix D

$$(57) \quad D^2 w^p$$

- Determine the new descent direction by orthogonalizing with respect to the $D^* D$ inner product the vector Dw^p and all the previously computed search directions

$$(58) \quad \gamma_j^p = -(D^2 w^p, \overline{Dw^j}) \quad \forall j = 0, \dots, p$$

$$(59) \quad w^{p+1} = Dw^p + \sum_{j=0}^p \gamma_j^p w^j$$

- Compute Dw^{p+1}

$$(60) \quad Dw^{p+1} = D^2 w^p + \sum_{j=0}^p \gamma_j^p Dw^j$$

- Normalize w^{p+1}

$$(61) \quad w^{p+1} = \frac{w^{p+1}}{\sqrt{(Dw^{p+1}, \overline{Dw^{p+1}})}} \quad Dw^{p+1} = \frac{Dw^{p+1}}{\sqrt{(Dw^{p+1}, \overline{Dw^{p+1}})}}$$

where (w, v) represents the scalar product of the vectors w and v in $L^2(\Gamma \times \Gamma)$, and \overline{w} the complex conjugate of the vector w . Using the properties of orthogonality of the different vectors, one can show that this algorithm converges with a number of iterations that is smaller or equal to the dimension of matrix D .

3.3.2. Cost and implementation issues. The main part of the computations is associated with the matrix-vector product, for which one has to solve a Helmholtz problem with radiation conditions at each iteration, and in each subdomain. The remainder of the computation consists of scalar products and linear combinations of vectors.

As stated previously, the product of a vector by matrix D needs only data that is local to each processor. This product is performed by first using matrices A_i^{-1} , M_{ij} and R_{ij} which are local to the subdomains, and then assembling the result over all the subdomains.

The fact that A_i is a symmetric matrix allows us to use a Crout factorization so that the products by matrix A_i^{-1} can be obtained by forward and backward substitutions. Since matrix R_{ij} is a restriction matrix, it does not need to be stored. Furthermore, the rank of matrix M_{ij} is equal to the number of degrees of freedom on the interface between two subdomains. The use of a direct local solver, and the properties of the operator on the interface ensure that the proposed domain decomposition method has good convergence properties and is more robust than other iterative methods. Also note that because only local matrices are factored, this method is more economical than direct ones.

From the implementation viewpoint, the assembly part requires exchanging messages containing one-dimensional arrays defined on the interfaces such as descent directions, or Lagrange multipliers. Therefore, the amount of data exchanged

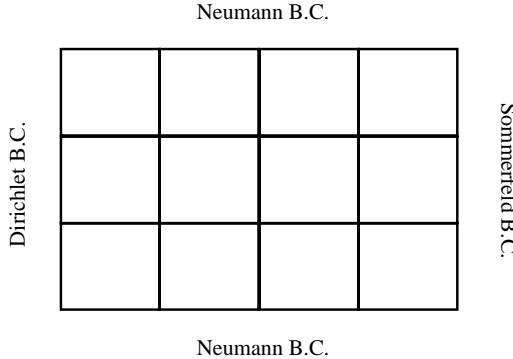


FIGURE 1. Problem definition

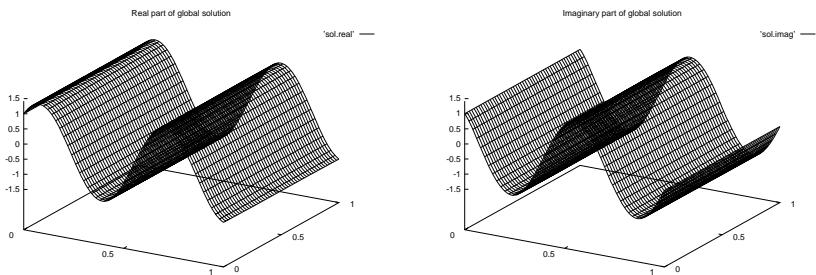


FIGURE 2. Real and imaginary parts of the exact solution

is small compared to the amount of computations performed. In other terms, the product of a vector by matrix D consists in local forward and backward substitutions followed by data exchanges between processors, and therefore the proposed method is easily parallelizable on any multiprocessor.

4. Numerical scalability analysis

Here, we perform a set of numerical experiments to assess the convergence of the proposed method for an increasing problem size, and/or an increasing number of subdomains, and/or an increasing wavenumber. More specifically, we consider a two-dimensional rectangular waveguide problem with a uniform source located on the west side, and reflecting boundaries at the north and south sides. The exact solution of this sharp problem is a plane wave traveling from west to east. The domain Ω is discretized by finite elements and partitioned in a number of subdomains. Homogeneous Neumann boundary conditions are applied on the north and south sides of the domain, a non-homogeneous Dirichlet condition is applied on its west side, and an absorbing condition on its east side.

The geometry of the domain and the real and imaginary parts of the exact solution are shown on Fig. 1 and Fig. 2.

First, we investigate the dependence of the convergence of the proposed method on the mesh size (Fig. 3). For this purpose, we fix the wavenumber k and the number of subdomains and consider a series of mesh sizes $h, h/2, h/3, \dots$. The

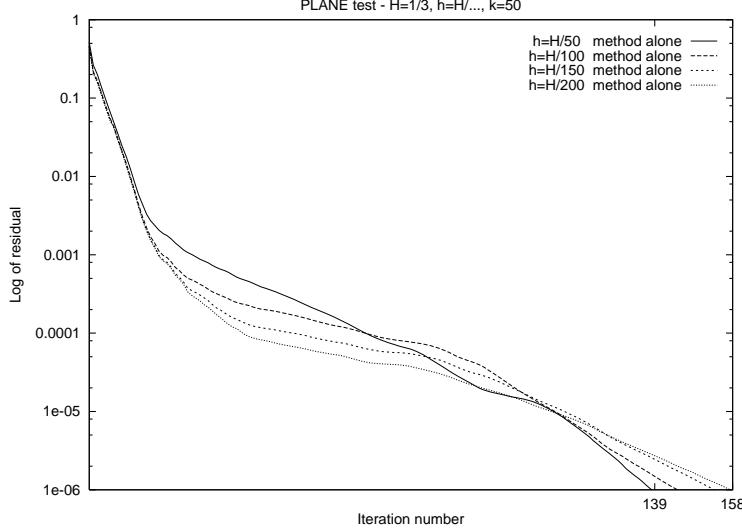


FIGURE 3. Effect on convergence of the mesh size

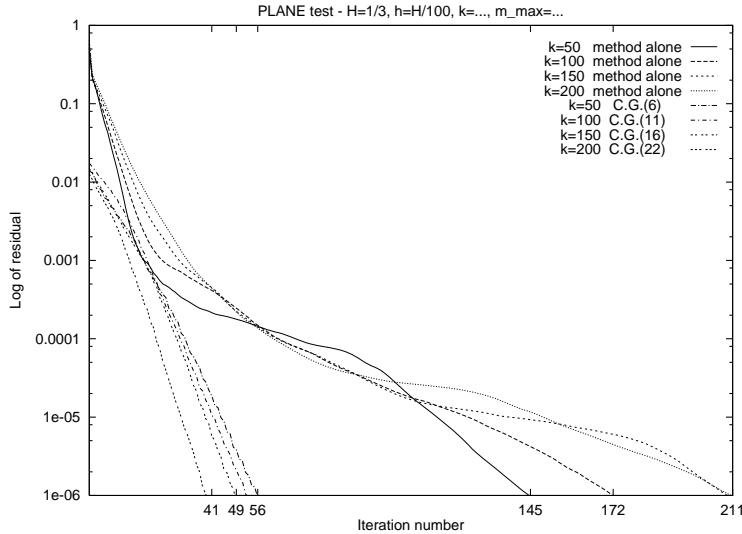


FIGURE 4. Effect on convergence of the wavenumber

results reported in Figure 3 show that the convergence of the proposed domain decomposition method is only weakly dependent on the mesh size h , which is quite impressive given that no preconditioner has yet been explicitly introduced. This corroborates the fact that operator Q_{ij} is unitary.

The next figure depicts the variation of the convergence of the method with respect to the wavenumber for a fixed mesh size and a fixed number of subdomains (Figure 4). The results reported in Figure 4 reveal a sublinear dependence on the wavenumber k . Practically, this indicates that when the frequency of the problem is increased, the convergence of the method deteriorates.

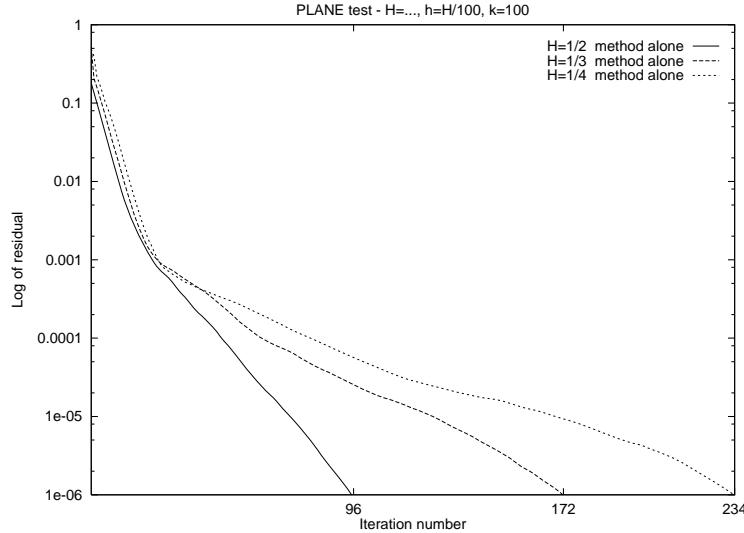


FIGURE 5. Effect on convergence of the number of subdomains

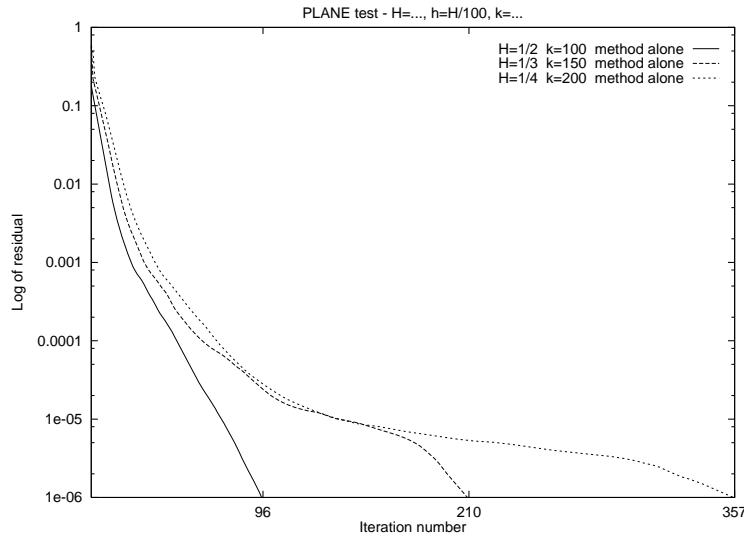


FIGURE 6. Convergence with a fixed number of wavelengths per subdomain

In order to study the effect of the number of subdomains, we fix the number of degrees of freedom in the local meshes and the wavenumber, and increase the number of subdomains. The corresponding results (Figure 5) show a linear dependence with respect to the number of subdomains. The last study (Figure 6) shows the effect on convergence of a simultaneous variation of the wavenumber and the number of subdomains where kH is kept constant, H being the mean diameter of a subdomain. The results clearly show a linear dependence of convergence on kH .

In summary, the method as presented so far seems to scale with the problem size (mesh size), but not with the wavenumber and/or the number of subdomains.

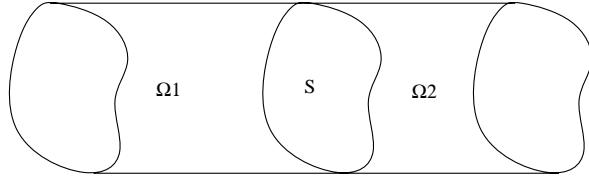


FIGURE 7. Geometry of the model problem

Therefore, our next objective is to develop preconditioning techniques for ensuring scalability with respect to the number of subdomains, and improving convergence in the high frequency regime.

5. Preconditioning techniques

5.1. Definitions. Here, we restrict our analysis to the following canonical problem (cf Figure 7).

Let S be a bounded set of \mathbb{R}^2 with a regular surface. Domain Ω under consideration is $S \times [0, L]$. We use two subdomains Ω_1 and Ω_2 defined by

$$(62) \quad \Omega_1 = S \times [0, L_1]$$

$$(63) \quad \Omega_2 = S \times [L_1, L_1 + L_2]$$

with $0 < L_1$, $0 < L_2$ and $L_1 + L_2 = L$. The sectional variables are denoted by x, y , the fiber variable is denoted by z . We will also use $(x, y, z) = (Y, z) = X$. We denote by $\Gamma_I = S \times L_1$ the interface between Ω_1 and Ω_2 .

The interesting point about this problem is that it separates the variables and facilitates the explanation of some features of the method.

Using the previously introduced notation, the matrix D of the condensed problem can be written as

$$(64) \quad D = \begin{bmatrix} I & Q_2 \\ Q_1 & I \end{bmatrix}$$

where Q_1 (resp. Q_2) is a discrete form of the unitary operator introduced in the notation (29), associated with subdomain Ω_1 (resp. Ω_2). The spectrum of matrix D spreads on the unit circle of the complex plane centered in 1 and has two accumulation points : one in 0 and the other in 2 (cf. Fig. 8).

REMARK 12. If the existence of areas of accumulation of eigenvalues generally accelerates the convergence of Krylov-like methods, it is nonetheless clear that the accumulation point located in 0 deteriorates the conditioning of matrix D .

REMARK 13. Increasing the frequency for a given mesh has two effects. First, it diminishes the density around the accumulation points and increases the dispersion of the spectrum. Second, the numerical accumulation points move away from 0 and 2. These two properties have contradictory effects on the convergence speed and explain the various crossings of the convergence curves.

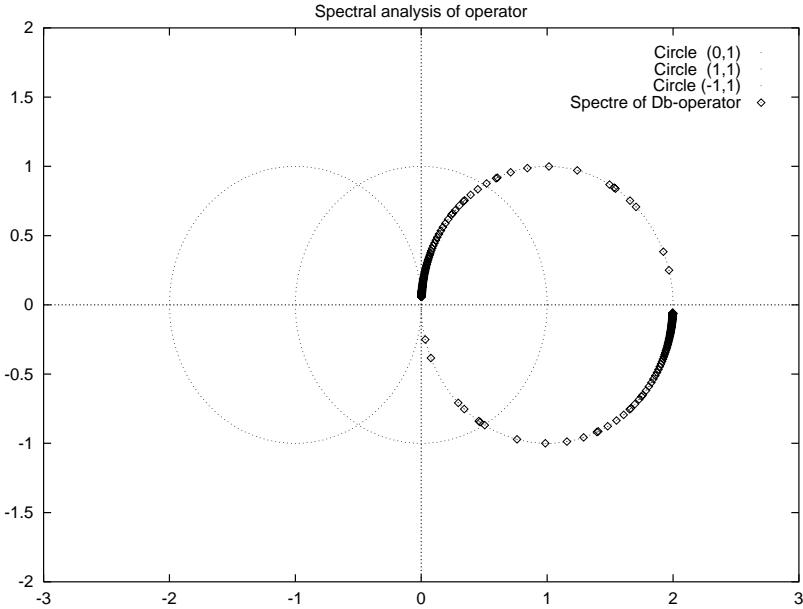


FIGURE 8. Spectral analysis

5.2. Spectral analysis. When considering our canonical problem, the solutions of the Helmholtz problem in Ω_1 and Ω_2 may be decomposed as the sums of

$$(65) \quad \psi_n(z)\phi_n(Y)$$

where ϕ_n are the eigenfunctions of the Laplace operator on S denoted by Δ_S . Hence, we have

$$(66) \quad \Delta_S\phi_n = -(\lambda_n)^2\phi_n.$$

Thus, the functions ψ_n in each of the two subdomains satisfy the ordinary differential equation

$$(67) \quad \psi_n'' + [k^2 - (\lambda_n)^2]\psi_n = 0.$$

Two cases must be distinguished

- $k^2 > (\lambda_n)^2$: this is the propagative case. We denote $k'_n = \sqrt{k^2 - (\lambda_n)^2}$ and we have $\psi_n(z) = ae^{ik'_n z} + be^{ik'_n z}$.
- $k^2 > (\lambda_n)^2$: this is the case of a vanishing wave. We denote $k'_n = \sqrt{-k^2 + (\lambda_n)^2}$ and we have $\psi_n(z) = ae^{k'_n z} + be^{k'_n z}$.

In each case, a and b are determined from the boundary conditions at $z = 0$ or $z = L$.

Let us interpret physically the above two cases. The first case corresponds to a wave which oscillates slowly on the interface and propagates through the subdomains. The second case corresponds to a wave which oscillates so rapidly on the interface that it cannot propagate in the subdomains. In order to speed up convergence, we develop two complementary preconditioners that target the two different cases. The high frequency phenomena on the interface — which corresponds to vanishing waves in the subdomains and hence local waves around the interfaces

— will be filtered by a local preconditioner. We will construct the local preconditioner as an approximate inverse of the hermitian part of D . The low frequency phenomena — which corresponds to waves propagating in all the subdomains — will be filtered by a global preconditioner that will be referred to in the sequel as the *Coarse Grid* preconditioner. This coarse grid preconditioner is a projection on the space orthogonal to a set of functions which are defined on the interfaces. Here, these functions are chosen as low frequency plane waves defined locally on each interface Γ_{ij} .

5.3. The local preconditioner.

5.3.1. *The principle.* Here, our goal is to replace the solution of the linear system $D\lambda = b$ by the solution of the system $MD\lambda = Mb$, where matrix M is an approximate inverse of matrix $(D + D^*)/2$. We denote by n the vector normal to Γ_I and pointing towards the increasing z . Thus,

$$(68) \quad Q_1 : \frac{\partial u^1}{\partial n} + iku^1 \mapsto \frac{\partial u^1}{\partial n} - iku^1$$

$$(69) \quad Q_2 : -\frac{\partial u^2}{\partial n} + iku^2 \mapsto -\frac{\partial u^2}{\partial n} - iku^2$$

Let us represent Q_1 (resp. Q_2) in the basis of the functions $\phi_n(Y)$.

If $u^1|_{\Gamma_I} = \phi_n(Y)$, then, in Ω_1 ,

$$u^1 = \frac{\psi_n(z)}{\psi_n(L_1)} \phi_n(Y),$$

and

$$\partial_n u^1 = \frac{\psi'_n(L_1)}{\psi_n(L_1)} \phi_n(Y).$$

Hence,

$$(70) \quad Q_1 : \phi_n(Y) \mapsto \frac{\psi'_n(L_1) - ik\psi_n^1(L_1)}{\psi'_n(L_1) + ik\psi_n^1(L_1)} \phi_n(Y).$$

Similarly,

$$(71) \quad Q_2 : \phi_n(Y) \mapsto \frac{-\psi'_n(L_2) - ik\psi_n^2(L_2)}{-\psi'_n(L_2) + ik\psi_n^2(L_2)} \phi_n(Y).$$

In the case of Dirichlet boundary conditions on the two faces of the cylinder, one has

- $\psi_n^1(z) = \sin(k'_n z)$ in the propagative case,
- $\psi_n^1(z) = \text{sh}(k'_n z)$ in the vanishing case,
- $\psi_n^2(z) = \sin(k'_n(z - L))$ in the propagative case,
- $\psi_n^2(z) = \text{sh}(k'_n(z - L))$ in the vanishing case.

Hence,

- $\frac{\psi'_n}{\psi_n^1}(L_1) = k'_n \cotg(k'_n L_1)$ (prop. case),
- $\frac{\psi'_n}{\psi_n^1}(L_1) = k'_n \coth(k'_n L_1)$ (van. case),
- $\frac{\psi'_n}{\psi_n^2}(L_2) = -k'_n \cotg(k'_n L_2)$ (prop. case),

$$\bullet \frac{\psi_n^{2'}}{\psi_n^2}(L_2) = -k'_n \coth(k'_n L_2) \text{ (van. case).}$$

It follows that, in the propagative case,

$$(72) \quad Q_1 : \phi_n \longrightarrow \frac{k'_n \cotg(k'_n L_1) - ik}{k'_n \cotg(k'_n L_1) + ik} \phi_n$$

$$(73) \quad Q_2 : \phi_n \longrightarrow \frac{k'_n \cotg(k'_n L_2) - ik}{k'_n \cotg(k'_n L_2) + ik} \phi_n$$

For the vanishing case, one has to turn the cotangent functions into hyperbolic cotangent functions.

Let us denote by λ_n^1 and λ_n^2 the eigenvalues of Q_1 and Q_2 that have appeared above. In the ϕ_n basis, operator D can be written as

$$(74) \quad D = \begin{bmatrix} 1 & \lambda_n^2 \\ \lambda_n^1 & 1 \end{bmatrix}$$

and its hermitian part, denoted by HD , is

$$(75) \quad HD = \begin{bmatrix} 1 & (\lambda_n^2 + \bar{\lambda}_n^1)/2 \\ (\lambda_n^1 + \bar{\lambda}_n^2)/2 & 1 \end{bmatrix}.$$

To filter the vanishing modes, that is, the modes having a spatial frequency on the interface that is greater than the wavenumber of the problem, we are going to look for an approximate inverse of the hermitian part of matrix D in the limit $\lambda_n \gg k$.

In this limit, we have : $\coth(k'_n L_i) = 1 + \mathcal{O}(1/\lambda_n^p)$, $\forall p$.
Hence,

$$(76) \quad \lambda_n^1 = \frac{(\lambda_n)^2 - 2ik\sqrt{(\lambda_n)^2 - k^2} - 2k^2}{(\lambda_n)^2}$$

and,

$$(77) \quad \lambda_n^2 = \frac{(\lambda_n)^2 - 2ik\sqrt{(\lambda_n)^2 - k^2} - 2k^2}{(\lambda_n)^2}$$

One can deduce

$$(78) \quad (\lambda_n^1 + \bar{\lambda}_n^2)/2 = 1 - 2\frac{k^2}{(\lambda_n)^2} + \mathcal{O}(1/\lambda_n^p), \forall p.$$

Similarly,

$$(79) \quad (\bar{\lambda}_n^1 + \lambda_n^2)/2 = 1 - 2\frac{k^2}{(\lambda_n)^2} + \mathcal{O}(1/\lambda_n^p), \forall p.$$

The inverse of HD can be written as

$$(80) \quad (HD)^{-1} = \begin{bmatrix} 1 & -(\lambda_n^2 + \bar{\lambda}_n^1)/2 \\ -(\lambda_n^1 + \bar{\lambda}_n^2)/2 & 1 \end{bmatrix} \frac{1}{1 - |\frac{\lambda_n^2 + \bar{\lambda}_n^1}{2}|^2}.$$

which simplifies to

$$(81) \quad (HD)^{-1} = (1/4) \begin{bmatrix} 1 + \frac{(\lambda_n)^2}{k^2} & 1 - \frac{(\lambda_n)^2}{k^2} \\ 1 - \frac{(\lambda_n)^2}{k^2} & 1 + \frac{(\lambda_n)^2}{k^2} \end{bmatrix} + \mathcal{O}(1/\lambda_n^2).$$

Since the functions ϕ_n are eigenfunctions of Δ_S with $-(\lambda_n)^2$ as eigenvalues,

$$(82) \quad (HD)^{-1} = (1/4) \begin{bmatrix} 1 - \frac{\Delta_S}{k^2} & 1 + \frac{\Delta_S}{k^2} \\ 1 + \frac{\Delta_S}{k^2} & 1 - \frac{\Delta_S}{k^2} \end{bmatrix} + L$$

where L is a continuous operator from Sobolev space H^s into H^{s+2} . We will use the first part of the formula defining $(HD)^{-1}$ as a local preconditioner.

The goal of this local preconditioner is to filter the eigenmodes associated with the eigenvalues that are close to zero. But, as a drawback, it also changes the behavior of the other modes, and principally the low frequency ones.

Hence, we will have to correct this drawback by designing a preconditioner that is associated with the propagative modes in order to achieve a good convergence. This will be done by our coarse grid preconditioner that we present and discuss later in this paper.

5.3.2. The Preconditioned Generalized Conjugated Residual algorithm. In the case where a preconditioner M of matrix D is known, one can use a modified *Generalized Conjugated Residual* algorithm to solve the linear system $D\lambda = b$. In this algorithm, the successive descent direction vectors are built in order to create a D^*D orthogonal basis of the successive Krylov spaces

$$K_{p+1} = \{Mg_0, MDMg_0, \dots, (MD)^p Mg_0\}.$$

The algorithm is now presented in details.

- Initialization

$$(83) \quad \lambda_0, \quad g_0 = D\lambda_0 - b, \quad w^0 = Mg_0$$

- Computation of the product of vector w^0 by matrix D then normalization

$$(84) \quad Dw^0 / (Dw^0, \overline{Dw^0})^{1/2}$$

- Iteration $p + 1$ of *Preconditioned generalized conjugated residuals algorithm* for $p \geq 0$
 - Determination of the optimal descent coefficient

$$(85) \quad \rho^p = -(g^p, \overline{Dw^p})$$

- Update of the solution and its residual

$$(86) \quad \lambda^{p+1} = \lambda^p + \rho^p w^p$$

$$(87) \quad g^{p+1} = g^p + \rho^p Dw^p$$

- Computation of the product of vector Dw^p by matrix M

$$(88) \quad MDw^p$$

- Computation of the product of MDw^p by matrix D

$$(89) \quad DMDw^p$$

- Determination of the new descent direction by orthogonalizing for D^*D vector MDw^p with respect to the previously computed directions

$$(90) \quad \gamma_j^p = -(DMDw^p, \overline{Dw^j}) \quad \forall j = 0, \dots, p$$

$$(91) \quad w^{p+1} = M D w^p + \sum_{j=0}^p \gamma_j^p w^j$$

– Determination of vector Dw^{p+1}

$$(92) \quad Dw^{p+1} = DMDw^p + \sum_{j=0}^p \gamma_j^p Dw^j$$

– Normalization of the new descent direction

$$(93) \quad w^{p+1} = \frac{w^{p+1}}{\sqrt{(Dw^{p+1}, Dw^{p+1})}} \quad Dw^{p+1} = \frac{Dw^{p+1}}{\sqrt{(Dw^{p+1}, Dw^{p+1})}}$$

5.3.3. *Cost issues.* The local preconditioner described above requires only a matrix vector multiplication on the subdomain interfaces, and therefore is economical and parallelizable. It is reminiscent of the “lumped” preconditioner for elasticity problems [10].

5.4. The coarse grid preconditioner.

5.4.1. *The principle.* The goal of this method which was first introduced by Farhat, Chen, and Mandel in [5] for time-dependent elasticity problems is to build an $n+1$ dimensional space W for a space V called *Coarse Grid*, and then to perform the iterations of GCR in a space orthogonal to W . With a good choice of the basis functions of the coarse grid, we aim at a better convergence of the algorithm, since it starts with the initial knowledge of $n+1$ descent directions. From what has been shown above, we choose for V , the space spanned by $\{v_0, \dots, v_m\}$ where the v_i are low frequency functions on an interface. This space will filter the eigenmodes associated with propagative phenomena.

More details on the theory of the coarse grid preconditioner can be found in [6]. In this paper, we implement the coarse grid preconditioner by means of reconjugations within the GCR algorithm. Of course it is not the unique way to do this and we could have followed a method using the definition of matrix operators as presented in [5] for instance, but the results would not have been changed.

In order to use the projected GCR algorithm, one has to build a basis $W = \{w_0, \dots, w_n\}$ that is D^*D orthonormal from basis V . As matrix D is regular, if vectors v_i are linearly independent, it will be the same for vectors w_i and therefore $m = n$. We want to construct W so that

$$w_i = \sum_{j=0}^m h_{ji} v_j \quad \text{and} \quad (Dw_i, \overline{Dw_j}) = 0 \quad \forall i \neq j$$

In matrix notation we have

$$(94) \quad W = VH \quad \text{such that} \quad (DW, \overline{DW}) = I$$

This approach produces the same effect as a *QR* factorization of basis V . Let us consider a Cholesky decomposition LL^* of matrix $(DV)^*(DV)$. Then the equality (94) becomes

$$(DVH)^*(DVH) = H^* LL^* H = I$$

By identifying factors, we deduce : $H = L^{-*}$. Hence, the computation of the basis W simply amounts to forward substitutions in the system $WL^* = V$. Once all the vectors of W are computed, we can apply the Projected GCR algorithm described next.

5.4.2. *The projected GCR algorithm.* When a D^*D orthonormal basis $\{w^0, \dots, w^n\}$ is known, it is possible to use a modified GCR, where the modification amounts to building the successive descent directions by reconjugating them with vectors $\{w^0, \dots, w^n\}$. The successive steps of the algorithm are

- Initialization

$$(95) \quad \tilde{\lambda}_0, \quad \tilde{g}_0 = D\tilde{\lambda}_0 - b$$

- Re-initialization by reconjugation

$$(96) \quad \rho_j = (g^0, \overline{Dw^j}) \quad \forall j = 0, \dots, n$$

$$(97) \quad \lambda_0 = \tilde{\lambda}_0 + \sum_{j=0}^n \rho_j w^j, \quad g_0 = \tilde{g}_0 + \sum_{j=0}^n \rho_j Dw^j$$

- Computation of the product of vector g_0 by matrix D

$$(98) \quad Dg_0$$

- Computation of the first descent direction

$$(99) \quad \gamma_j = -(g_0, \overline{Dw^j}) \quad \forall j = 0, \dots, n$$

$$(100) \quad w^{n+1} = g_0 + \sum_{j=0}^n \gamma_j w^j$$

- Determination of the quantity Dw^{n+1}

$$(101) \quad Dw^{n+1} = Dg_0 + \sum_{j=0}^n \gamma_j Dw^j$$

- Normalization of the first descent direction

$$(102) \quad w^{n+1} = \frac{w^{n+1}}{\sqrt{(Dw^{n+1}, \overline{Dw^{n+1}})}} \quad Dw^{n+1} = \frac{Dw^{n+1}}{\sqrt{(Dw^{n+1}, \overline{Dw^{n+1}})}}$$

- Iteration $p + 1$ of the projected GCR for $p \geq n + 1$

- Determination of the optimal descent coefficient

$$(103) \quad \rho^p = -(g^p, \overline{Dw^p})$$

- Update of the solution and the residual

$$(104) \quad \lambda^{p+1} = \lambda^p + \rho^p w^p$$

$$(105) \quad g^{p+1} = g^p + \rho^p Dw^p$$

- Computation of the product of vector Dw^p by matrix D

$$(106) \quad D^2w^p$$

- Determination of the new descent direction by orthogonalizing for D^*D vector MDw^p with respect to the previously computed directions

$$(107) \quad \gamma_j^p = -(D^2w^p, \overline{Dw^j}) \quad \forall j = 0, \dots, p$$

$$(108) \quad w^{p+1} = Dw^p + \sum_{j=0}^p \gamma_j^p w^j$$

– Determination of vector Dw^{p+1}

$$(109) \quad Dw^{p+1} = D^2 w^p + \sum_{j=0}^p \gamma_j^p Dw^j$$

– Normalization of the descent direction vector

$$(110) \quad w^{p+1} = \frac{w^{p+1}}{\sqrt{(Dw^{p+1}, Dw^{p+1})}} \quad Dw^{p+1} = \frac{Dw^{p+1}}{\sqrt{(Dw^{p+1}, Dw^{p+1})}}$$

5.4.3. Cost issues. Here again, the major part of the computation cost resides in the matrix-vector product. The reconjugations are just scalar products and linear combinations of vectors. It follows that the computation of basis $W = \{w_0, \dots, w_n\}$, obtained by D^*D orthonormalization of basis V represents an important amount of computation. It would be as expensive as the computation of the $n + 1$ first directions of GCR, if basis V were not defined locally, interface by interface. The choice made for basis V and the fact that the Lagrange multipliers are defined on the double of the global interface restrict the computation of the product of a vector v_i by matrix D to a product by the local matrix A_i , which is a forward-backward substitution, and to exchanges of data between neighbor subdomains. The cost to build the basis W is associated with that of performing in each subdomain Ω_i as many forward-backward substitutions as the number of interfaces of the subdomain, and this for each coarse function defined on an interface. On a parallel processor, if the number of coarse grid functions per interface remains constant, an increase of the number of subdomains would have no effect on the local computational cost of basis W . The cost of the basis functions of the coarse grid is thus small compared to the cost of the computation of the $n + 1$ first directions of the GCR algorithm.

In the following section, we present the convergence curves when the preconditioners proposed here are used. These curves show that the performance of the proposed domain decomposition method equipped with the preconditioners described here exhibits a low dependency on the wavenumber and on the number of subdomains.

6. Performance of the preconditioned domain decomposition method

Here, we assess the impact of the preconditioners presented in the previous sections on the convergence of the proposed domain decomposition method. For this purpose, we employ the same test problem as that introduced in Section 4.

In a first step we investigate the influence of the number of coarse grid basis functions on the number of iterations (Figure 9). As stated before, the coarse grid preconditioner aims at filtering the phenomena associated with low frequencies on the interfaces. If we choose a number of coarse grid basis functions that is too small compared to $m_{max} = k\pi/L$, some low frequency modes will be damped but the others will slow down the method (L is the length of the interface). If we increase the number of basis functions per interface to exceed m_{max} , the coarse grid will affect the middle and high frequency oscillations on the interfaces. In other words, the global preconditioner will interfere with the local one. Hence, it seems more

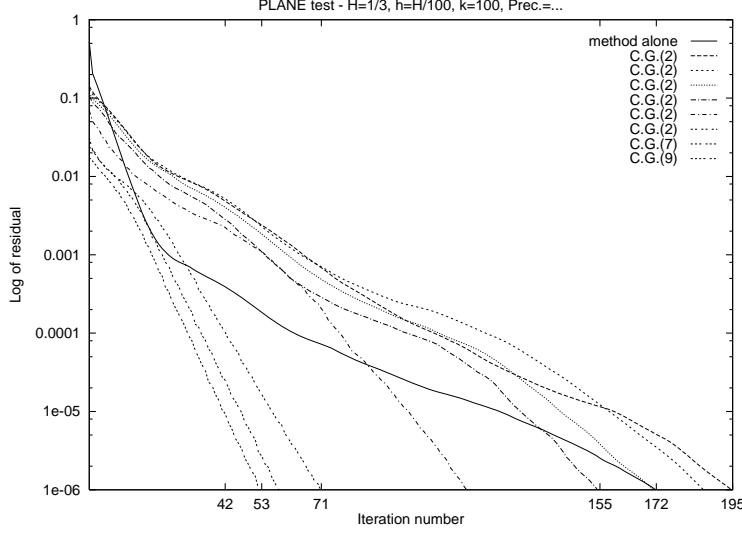


FIGURE 9. Effect of the size of the coarse problem

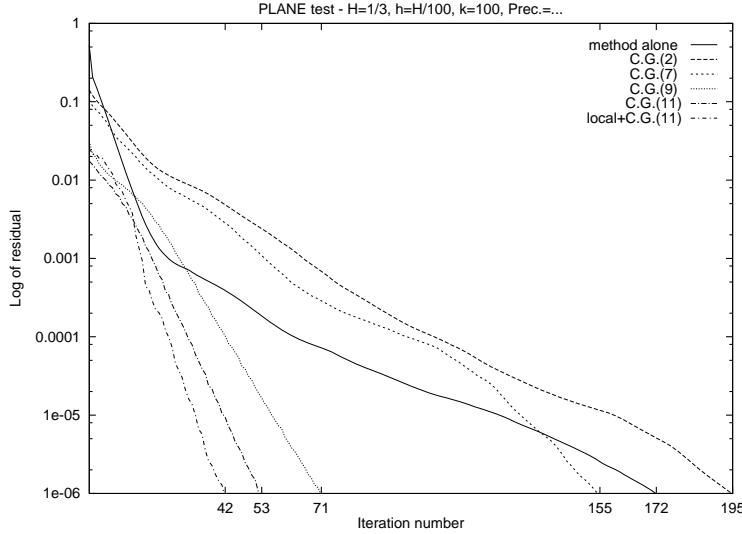


FIGURE 10. Effect of the local preconditioner

sensible to use the coarse grid preconditioner only for low frequencies and to use the local preconditioner for high frequency phenomena. This strategy is also justified by the fact that the local preconditioner increases the granularity of the method without adding reconjugations at each iteration.

In Figure 10, one can see that the local preconditioner accelerates convergence once the low frequency modes have been filtered.

In Section 4, we have exhibited the weak dependency of our method on the mesh size. Here, we show in Figure 11 that this dependency is even weaker when the two preconditioners are used.

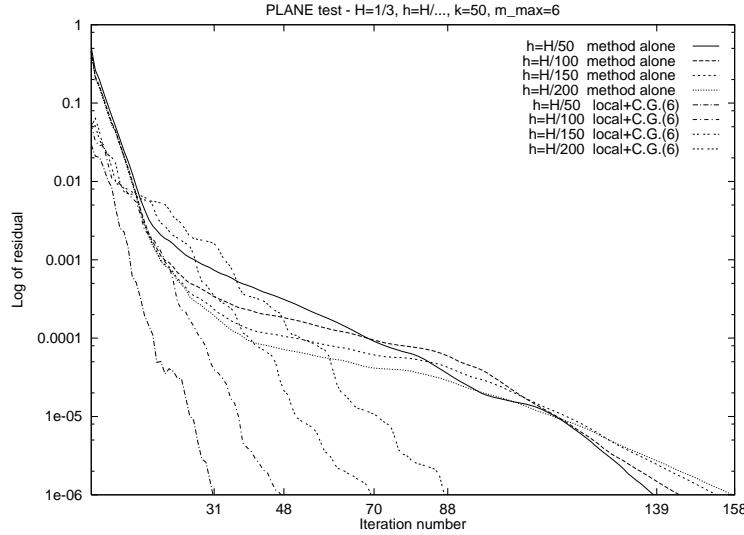


FIGURE 11. Optimal convergence results

Next, we fix the size of the local meshes and the geometry of the global problem and focus on the value of parameter m_{max} , where m_{max} denotes the largest integer satisfying $m\pi < kH$ where H is the length of a subdomain interface. In other words, all the functions $\phi_m(Y)$ with $m > m_{max}$ correspond to vanishing modes that do not propagate on a long range. We consider a given frequency and a given number of subdomains N . If N is increased, the mean diameter H of the subdomains is reduced, and so is m_{max} . The total number of basis functions of the coarse grid space remains unchanged, but, locally the number of basis functions per subdomain decreases; hence, the computational cost is reduced. Figure 12 shows that the coarse grid preconditioner achieves a weak dependence of the method on the the number of subdomains.

Next, we stress that the number of coarse grid functions must vary with the wavenumber k . Indeed, when the frequency is increased, one has to proportionally increase the number m_{max} in order to filter all the propagative modes. Figure 13 shows that, with an appropriate variation of the number of these functions per interface, it is possible to increase the frequency with only a small variation in the convergence histories. If the wavelength diminishes proportionally to the size of the subdomain, i.e. so that the product kH remains constant, one can see that the number m_{max} remains constant. In other terms, in that case one does not have to increase the number of coarse grid functions for each subdomain interface boundary (see Figure 14). This property is important for realistic exterior acoustics problems.

In summary, the results reported herein show that the performance of the proposed domain decomposition method equipped with the proposed local and global preconditioners is weakly dependent on the mesh size, the subdomain size, and the frequency of the problem, which makes this method uniquely efficient at solving high frequency exterior acoustics problems.

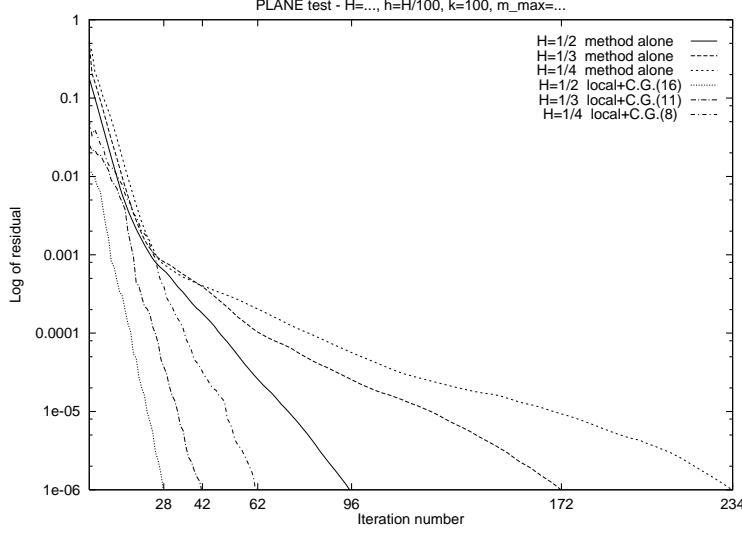


FIGURE 12. Effect of the coarse grid preconditioner

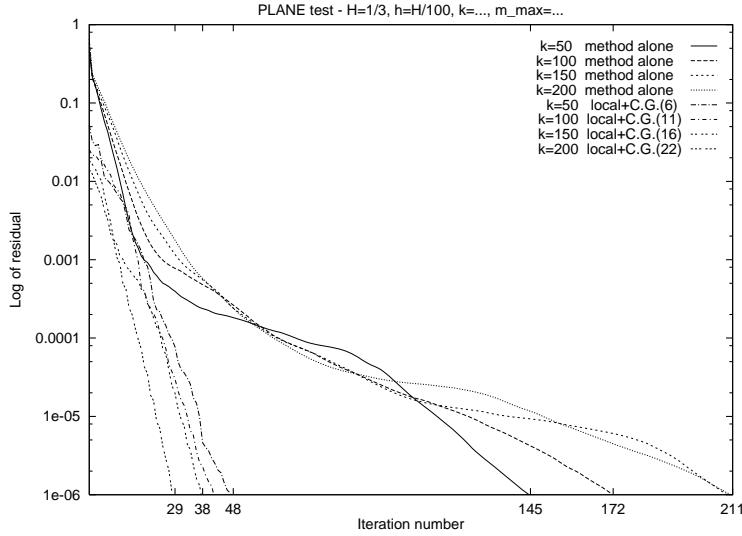


FIGURE 13. Performance results for the high frequency regime

7. Conclusions

In this paper, we have presented a Lagrange multiplier based domain decomposition method for solving iteratively large-scale systems of equations arising from the finite element discretization of high-frequency exterior Helmholtz problems. The proposed method relies on three key ideas: (1) the elimination of local resonance via the stabilization of each subdomain operator by a complex interface mass matrix associated with intersubdomain radiation conditions, (2) the use of a carefully constructed local preconditioner for filtering high frequency errors and accelerating convergence in the presence of fine meshes, and (3) the use of a global

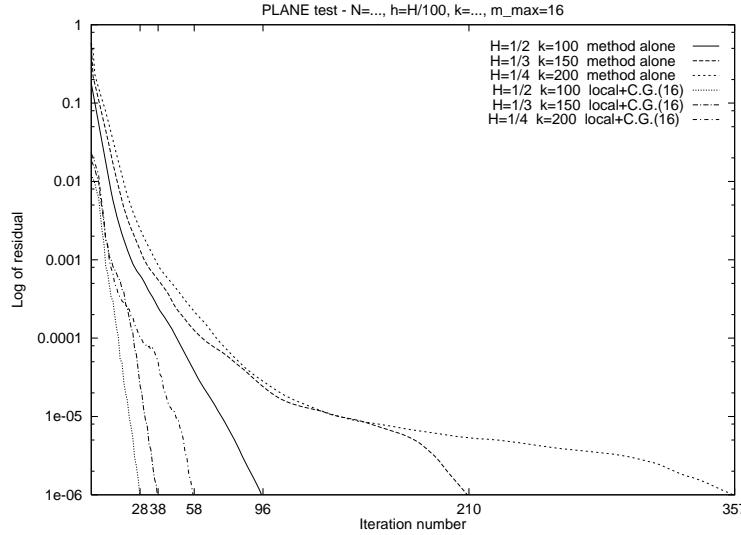


FIGURE 14. Effect of the coarse grid size

preconditioner constructed using a coarsening theory for filtering low frequency errors and accelerating convergence in the presence of fine mesh partitions. A unique characteristic of the proposed method is that, even in the absence of both preconditioners, its number of iterations grows at most polylogarithmically with the number of elements per substructure, and grows sublinearly with the wave number. Furthermore, when equipped with both the local and global preconditioners derived in this paper, the performance of the proposed method becomes almost insensitive to the frequency range and number of subdomains, which makes this method uniquely efficient at solving high frequency exterior acoustics problems [5].

8. Acknowledgements

The US authors acknowledge the support by the Office of Naval Research under Grant N-00014-95-1-0663.

References

1. A. de la Bourdonnaye, *Some formulations coupling volumic and integral equation methods for Helmholtz equation and electromagnetism*, Numerische Mathematik **69** (1995), 257–268.
2. ———, *A substructuring method for a harmonic wave propagation problem : Analysis of the condition number of the problem on the interfaces*, rapport de recherche No 95-35, CERMICS, France, 1995.
3. B. Després, *Décomposition de domaine et problème de Helmholtz*, Compte Rendu de l'Académie des Sciences **No 311 (Série I)** (1990), 313–316.
4. B. Després, *Domain decomposition method and the Helmholtz problem*, Mathematical and Numerical aspects of wave propagation phenomena, France, Strasbourg, 1991, pp. 44–52.
5. C. Farhat, P. S. Chen, and J. Mandel, *Scalable Lagrange multiplier based domain decomposition method for time-dependent problems*, Int. J. Numer. Meth. Engrg. **38** (1995), 3831–3853.
6. C. Farhat, P.-S. Chen, F. Risler, and F.-X. Roux, *A simple and unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers*, International Journal for Numerical Methods in Engineering, in press.
7. C. Farhat, A. Macedo, and M. Lesoinne, *The FETI-H method for the solution of high-frequency exterior Helmholtz problems*, submitted to Comput. Maths. Appl. Mech. Engrg.

8. C. Farhat, A. Macedo, F. Magoulès, and F.-X. Roux, *A Lagrange multiplier based domain decomposition method for the exterior Helmholtz problem*, Proceedings Fourth U.S. National Congress on Computational Mechanics, San Francisco, California, August 6-8, 1997.
9. C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Meths. Engrg. **Vol. 32** (1991), 1205–1227.
10. Charbel Farhat and François-Xavier Roux, *Implicit parallel processing in structural mechanics*, Computational Mechanics Advances (J. Tinsley Oden, ed.), vol. 2 (1), North-Holland, 1994, pp. 1–124.
11. C. Lacour and Y. Maday, *Two different approaches for matching nonconforming grids : the mortar element method and the feti method*, B.I.T. (1997).
12. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances **2** (1994), 121–220.
13. F.-X. Roux, *Méthode de décomposition de domaine pour des problèmes elliptiques*, Revue Calculateurs Parallèles, Volume 7, No 1, June 1994.

CERMICS, INRIA SOPHIA ANTIPOLIS, FRANCE
E-mail address: armel.de.La.bourdonnaye@sophia.inria.fr

UNIVERSITY OF COLORADO, DEPARTMENT OF AEROSPACE ENGINEERING SCIENCES, CAMPUS
 Box 429, BOULDER CO 80309-0429
E-mail address: charbel@alexandra.Colorado.edu

UNIVERSITY OF COLORADO, DEPARTMENT OF AEROSPACE ENGINEERING SCIENCES, CAMPUS
 Box 429, BOULDER CO 80309-0429
E-mail address: macedo@alexandra.Colorado.edu

ONERA, DIRECTION DE L'INFORMATIQUE, 29 Av. DE LA DIVISION LECLERC, BP72 92322
 CHATILLON CEDEX, FRANCE
E-mail address: magoules@onera.fr

ONERA, DIRECTION DE L'INFORMATIQUE, 29 Av. DE LA DIVISION LECLERC, BP72 92322
 CHATILLON CEDEX, FRANCE
E-mail address: roux@onera.fr

An Agglomeration Multigrid Method for Unstructured Grids

Tony F. Chan, Jinchao Xu, and Ludmil Zikatanov

1. Introduction

A new agglomeration multigrid method is proposed in this paper for general unstructured grids. By a proper local agglomeration of finite elements, a *nested* sequence of finite dimensional subspaces are obtained by taking appropriate linear combinations of the basis functions from previous level of space. Our algorithm seems to be able to solve, for example, the Poisson equation discretized on any shape-regular finite element grids with nearly optimal complexity.

In this paper, we discuss a multilevel method applied to problems on general unstructured grids. We will describe an approach for designing a multilevel method for the solution of large systems of linear algebraic equations, arising from finite element discretizations on unstructured grids. Our interest will be focused on the performance of an agglomeration multigrid method for unstructured grids.

One approach of constructing coarse spaces is based on generating node-nested coarse grids, which are created by selecting subsets of a vertex set, retriangulating the subset, and using piecewise linear interpolation between the grids (see [8, 5]). This still provides an automatic way of generating coarse grids and faster implementations of the interpolation in $O(n)$ time. The drawback is that in three dimensions, retetrahedralization can be problematic.

Another effective coarsening strategy has been proposed by Bank and Xu [1]. It uses the geometrical coordinates of the fine grid and the derefinement algorithm is based on the specified patterns of fine grid elements. The interpolation between grids is done by interpolating each fine grid node using only 2 coarse grid nodes. As a consequence of that the fill-in in the coarse grid matrices is reasonably small. The hierarchy of spaces is defined by interpolating the basis.

Recently a new approach, known as auxiliary space method, was proposed by Xu [16]. In this method only one non-nested (auxiliary) grid is created and then all consecutive grids are nested. This can be done by using as auxiliary grid a uniform

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35J05.

The first author was supported in part by NSF ASC-9720257.

The second author was supported in part by NSF DMS94-03915-1 and NSF DMS-9706949 through Penn State, by NSF ASC-92-01266 through UCLA and NSF ASC-9720257.

The third author was supported in part by NSF DMS-9706949 through Penn State, NSF ASC-9720257 and also by Center for Computational Mathematics and Applications of Pennsylvania State University.

one and interpolating the values from the original grid there. For a uniform grid then, there is a natural hierarchy of coarse grids and spaces. Such a procedure leads to optimal multigrid methods in some applications.

One promising new coarsening techniques is based on the agglomeration technique (see Koobus, Lallemand and Dervieux [11]). Instead of constructing a proper coarse grid, neighboring fine grid elements are aggregated together to form macroelements. Since these aggregated regions are not standard finite elements, appropriate basis functions and interpolation operators must be constructed on them. An algebraic construction of aggregated coarse grid spaces has been investigated by Vaněk, Mandel, and Brezina [6] and Vaněk, Křížková [7]. Their approach uses a simple initial interpolation matrix, which might not be stable, and then this matrix is smoothed and stabilized by some basic relaxation schemes, e.g. Jacobi method.

The pure algebraic definition of the coarse spaces has the advantage that there is no need to use any geometrical information of the grid or of the shape of the grid and the kind of finite elements used. We refer to a paper by Ruge and Stuben [13] on algebraic multigrid. Recent developments in this direction have been made by Braess [2] and Reusken [12]. The main issue in using pure “black-box” algebraic derefinement is that the coarse grid operators usually become denser and it is not clear how to control their sparsity except in some special cases.

Another approach in the definition of coarse spaces, known as composite finite element method was recently investigated by Hackbusch and Sauter in [10]. This method gives coarse space constructions which result in only few degrees of freedom on the coarse grid, and yet can be applied to problems with complicated geometries.

In this paper, we will consider a new and rather simple technique for defining *nested* coarse spaces and the corresponding interpolation operators. Our earlier experience shows that the definition of the sparsity pattern of the transfer operators and the definition of these operators themselves is the most crucial point in designing multigrid algorithms for elliptic problems. In the present paper we propose a technique based on the graph-theoretical approach. Our goal is to construct a “coarse grid” using only the combinatorial (not the geometrical properties) of the graph of the underlying fine grid. This coarse grid is formed by groups of elements called aggregated macroelements. Using this approach a macroelement grid can be constructed for any unstructured finite element triangulation. We can implement our algorithm with or without any use of the nodal coordinates. Based on this macroelement partition, we propose an interpolation technique which uses only arithmetic average based on clearly determined coarse grid nodes. This leads to savings in storage and CPU time, when such scheme is implemented. In fact, to store the interpolation matrix we only need to store integers. Although rather simple, such type of interpolation leads to a multigrid algorithm with nearly optimal performance. Moreover the algorithm naturally recovers the structure of the natural coarse grids if the fine grid is obtained by structured refinement. Although we present only 2D algorithms we believe that it can be extended for 3D problems as well.

The rest of the paper is organized as follows. In section 2 we state the differential problem and briefly comment on the finite element discretization. In section 3 we give the definition of the standard V -cycle preconditioner. In section 4 we describe in detail the two level coarsening algorithm. In section 4.3 the interpolation between grids is defined. The multilevel implementation of the algorithm is given

in Section 4.4. The stability and approximation properties are investigated in Section 5 under rather mild assumptions on the geometry of the coarse grids. In Section 6 results of several numerical experiments are presented.

2. A model problem and discretization

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain with boundary $\Gamma = \Gamma_D \cup \Gamma_N$, where Γ_D is a closed subset of Γ with positive measure. We consider the following variational formulation of elliptic PDE: Find $U \in H_D^1(\Omega)$ such that

$$(1) \quad a(U, v) = F(v) \text{ for all } v \in H_D^1(\Omega),$$

where

$$(2) \quad a(U, v) = \int_{\Omega} \alpha(x) \nabla U \cdot \nabla v dx, \quad F(v) = \int_{\Omega} F(x) v dx.$$

Here $H_D^1(\Omega)$ as usual denotes the Sobolev space which contains functions which vanish on Γ_D with square integrable first derivatives. It is well-known that (1) is uniquely solvable if $\alpha(x)$ is a strictly positive scalar function and F is square integrable.

We consider a finite element space of continuous piecewise linear functions $M_h \subset H_D^1(\Omega)$ defined on a triangulation T_h of Ω . Then the corresponding finite element discretization of (2) is: Find $u_h \in M_h$ such that

$$(3) \quad a(u_h, v_h) = F(v_h) \text{ for all } v_h \in M_h.$$

The discretization results in a linear system of equations:

$$(4) \quad Au = f,$$

where A is a symmetric and positive definite matrix, f is the right hand side and the nodal values of the discrete solution u_h will be obtained in the vector u after solving the system (4).

3. Multigrid method

In this section, we introduce the notation related to the multigrid method, and we define the (1-1) V -cycle preconditioner.

Let us consider the following simple iteration scheme:

$$(5) \quad u^{\ell+1} = u^\ell + B_J(f - Au^\ell) \quad \ell = 1, 2, \dots,$$

where B_J is the V -cycle preconditioner B_J to be defined. We assume that we have given a nested sequence of subspaces $M_0 \subset \dots \subset M_{J-1} \subset M_J \equiv M_h$, with $\dim(M_k) = n_k$. We assume that the matrices A_k , $k = 0, \dots, J$, are stiffness matrices associated with M_k . We also assume that the interpolation operators I_{k-1}^k and the smoothing operators S_k are given.

In our case B_J will correspond to (1-1) V -cycle preconditioner. For given $g \in M_k$ we define $B_k g$ as follows:

ALGORITHM 1. [(1-1) V -cycle]

0. If $k = 0$ then $B_0g = A_0^{-1}g$
1. *Pre-smoothing:* $x^1 = S_k^T g$
2. *Coarse grid correction:*
 1. $q^0 = (I_{k-1}^k)^T(g - A_kx^1);$
 2. $q^1 = B_{k-1}q^0;$
 3. $x^2 = x^1 + I_{k-1}^k q^1;$
3. *Post-smoothing:* $B_k g = x^2 + S_k(g - A_k x^2).$

The practical definition of such a preconditioner in the case of unstructured grids will be our main goal in the next sections. We will define proper interpolation (prolongation) operators I_{k-1}^k , for $k = 1, \dots, J$ and the subspace M_{k-1} by interpolating the nodal basis in M_k . In order to have convergence of the iteration (5) independent of the mesh parameters, the subspaces have to satisfy certain stability and approximation properties, namely, that there exists a operator $\Pi_k : H^1(\Omega) \rightarrow M_k$ such that:

$$(6) \quad \|\Pi_k v\|_{1,\Omega} \leq C \|v\|_{1,\Omega},$$

$$(7) \quad \|v - \Pi_k v\|_{0,\Omega} \leq Ch|v|_{1,\Omega}, \quad \forall v \in H^1(\Omega).$$

We will comment on these properties of the agglomerated spaces in Section 5. Once the subspaces are defined, the V -cycle algorithm can be implemented in a straightforward fashion using as coarse grid matrices, given by $A_{k-1} = (I_{k-1}^k)^T A_k I_{k-1}^k$.

General discussions concerning the convergence of this type of method and its implementation can be found in the standard references, e.g. Bramble [3], Hackbusch [9], Xu [15].

4. Agglomerated macroelements

The main approach we will take in the construction of I_{k-1}^k will be first to define a coarse grid formed by macroelements (groups of triangles) and then interpolate locally within each macroelement. In this section, we will present an algorithm for the definition of the coarse grid consisting of macroelements. We first identify the *set of coarse grid nodes*. The interpolation from coarse grid to the fine grid will use the values at these nodes. As a next step, for a given node on the fine grid we have to define its ancestors on the coarse grid (i.e. the coarse grid nodes which will be used in the interpolation). These ancestors are determined by partitioning the fine grid into *agglomerated macroelements* (such macroelements can be seen on Fig. 1) which in some sense are analogue of the finite elements, because they have vertices which are precisely the coarse grid nodes, and their edges are formed by edges of the underlying fine grid.

4.1. Some basic graph theory. In this subsection we introduce some basic notation and definitions. Given a finite element triangulation T_h of Ω , we consider the corresponding graph, denoted by $G = (V, E)$, where V is the set of vertices (grid nodes) and E is the set of edges (boundaries of the triangles). In this definition, the concept of vertex and edge happen to be the same for the underlying triangulation and for the graph corresponding to the stiffness matrix. Associated with the graph G , we will form our coarse grid on the so called maximal independent set (MIS, for short) which is a set of vertices having the following two properties: any two vertices in this set are *independent* in the sense that they are not connected by an edge, and the set is *maximal* in the sense that an addition of any vertex to the set will invalidate the aforementioned independent property. The *graph distance* between

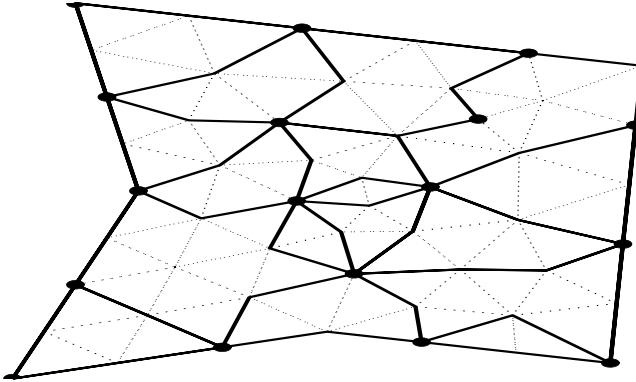


FIGURE 1. An example of macroelements

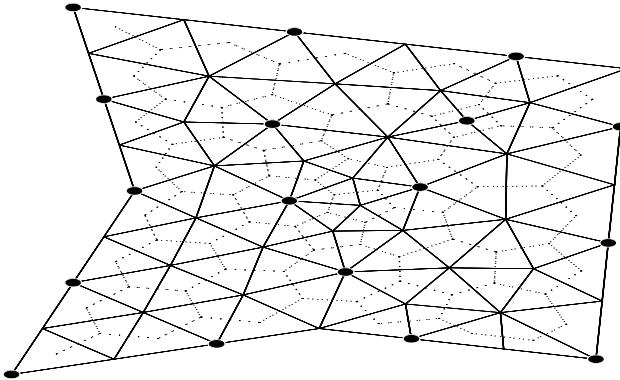


FIGURE 2. A triangulation and its dual mesh.

two vertices $v, w \in V$ is defined to be the length of the shortest path between these two vertices. A *matching* in G is any collection of edges such that no two edges in this collection share a vertex.

The construction of the macroelements will be based on the dual mesh (graph) of G defined as follows. Given a triangulation T_h and associated graph G , the dual graph $G' = (V', E')$ of G is:

- Each element $T \in T_h$ is a vertex in G' .
- Two vertices in G' are connected by an edge if and only if they share an edge in G , i.e. $(T_1, T_2) \in E'$ if and only if $T_1 \cap T_2 \in E$ (see Fig. 2).

4.2. Two level coarsening algorithm. In this section we describe in detail the heuristic algorithm for forming a coarse grid macroelements from a given finite element triangulation.

As a first step we define the set of coarse nodes to be a MIS in G . An MIS is obtained by a simple “greedy” (locally optimal) algorithm given as follows.

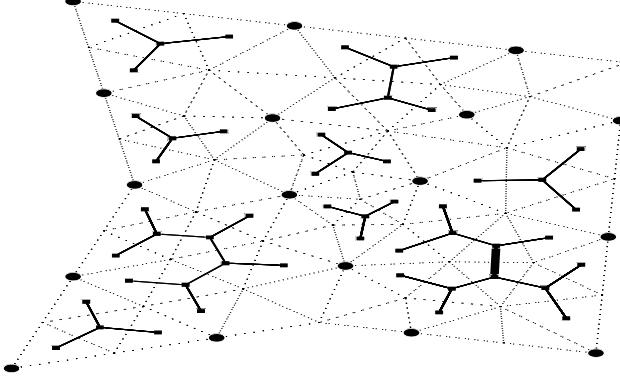


FIGURE 3. *Connected components in G^* .*

ALGORITHM 2 (MIS).

1. Pick an initial set of vertices V_0 (for example all boundary vertices).
2. Repeat:
 - (a) Apply a “greedy” algorithm to find MIS in V_0 .
 - (b) Mark all nodes at distance 1 from V_0 (here distance is the graph distance).
 - (c) Take as V_0 all vertices which are at distance 2 from V_0 and have not been explored (marked).
3. until $V_0 = \emptyset$.
4. Complete MIS by applying one step of “greedy” algorithm on V .

To define the macroelements, we use the fact that separating two triangles on the fine grid and putting them in different groups is equivalent to removing an edge in the dual graph G' .

We now describe how to form the initial partition of G into groups of elements.

For any coarse grid node k (i.e. $k \in MIS$) we pick the edges in G having this node as an end. To this set of edges $E_k \subset E$ corresponds a set $E'_k \subset E'$, namely E'_k contains exactly all edges between all $T \in T_h$ which have this particular coarse node as a vertex. As a first step we remove E'_k from E' . Applying this procedure for all coarse grid nodes results in a subgraph of G' , $G^* = (V^*, E^*)$ where $V^* = V'$ and $E^* = E' \setminus \cup_k E'_k$. The connected components in G^* will form the initial partition of Ω into groups of elements (see Fig. 3).

We note that there might be some isolated vertices in G^* and also some of the connected components might be considerably large. We first deal with the large groups (such a group can be seen on Fig. 3 in the right bottom corner of the domain) and we break them into smaller pieces. We consider a group of elements $M \subset T_h$ that corresponds to one connected component in G^* and denote the set of edges in M by E_M . We intend to break this group in pieces if there is an “interior” edge $e \in E_M$ such that $e \cap \partial M = \emptyset$. This breakup is done as follows (our considerations here are restricted only on $M \subset T_h$):

- From the subgraph formed by all edges $e \subset E_M$ such that $e \cap \partial M = \emptyset$, we form a matching. On the model grid (see Fig. 3) there is only one such edge in the whole domain.
- Remove the edges in the dual corresponding to the edges in the matching. In Fig. 3 this edge in the dual is drawn with thick line (near the right bottom corner of the domain). The pieces obtained by removing this edge are clearly seen on Fig. 1).

The situation with the isolated vertices in G^* is simpler. We propose two different ways of dealing with them as follows:

1. Since each isolated triangle (vertex in G^*) has as one vertex being a coarse grid point, the edge opposite to this vertex does not have a coarse grid node as an end, because our set of coarse grid nodes is a MIS. We group together two neighbors sharing this edge to form a macroelement. If such edge happens to be a boundary edge, we leave a single triangle to be a macroelement.
2. We group together all isolated neighbors. If such a group does not have more than 4 coarse grid vertices then we consider it as a new agglomerated macroelement. If it has more than 4 coarse grid vertices we proceed as in the previous step coupling triangles in this group two by two. In partitioning our model grid we have used precisely this way of grouping isolated triangles (see Fig 3, Fig. 1).

It is obvious that all triangles from the triangulation are either in a connected component in G^* or are isolated vertices in G^* . Thus we have explored all the triangles and every $T \in T_h$ is in some macroelement (see Fig. 1).

To summarize we give the following short description of the algorithm for agglomerating elements into macroelements:

ALGORITHM 3 (Coarse grid macroelements).

1. Identify coarse grid nodes by finding an MIS.
2. For any coarse node, remove all dual edges surrounding it.
3. Find connected components in the remaining dual graph. These connected components form most of the agglomerated regions.
4. Breakup “large” macroelements into smaller pieces.
5. Group the remaining triangles into contiguous groups as additional macroelements.
6. The remaining connected components in the dual are called “agglomerated elements”.

REMARK 4. Note that this algorithm will give a unique partition in agglomerated macroelements up to the choice of MIS and the edges in the matchings (if we need further breakup of large connected components in G^*).

We would like to elaborate a little more on the input data needed for the algorithm to work. The input we used was:

1. The grid (i.e. list of elements and correspondence “vertex–element”). From this correspondence we can easily define G in the usual way: two vertices are connected by an edge if and only if they share element.
2. The auxiliary graph G' whose vertices are the elements and the correspondence between edges in G and edges in G' .

Note that the algorithm we have described do not need the correspondence between edges in G and G' to be $(1 - 1)$ (as it is between the dual and primal graph). The only fact we used was: for a given edge in G the set of edges in G' which have to be removed is uniquely determined. This observation is important and will be used in the multilevel implementation of the algorithm.

4.3. The definition of coarse subspaces. In the present section we will describe a simple interpolation technique using the agglomerated macroelements. We also give a description how a multilevel variant of our derefinement algorithm can be implemented. With a grid agglomeration obtained as above, we need to define a coarse finite element space associated with the macroelements. This is equivalent to defining the interpolation between M_J and M_{J-1} . The interpolation is defined in the following way:

- Coarse nodes:
 - For the coarse nodes we simply define the interpolation to be the identity.
- Interior nodes:
 - For the nodes interior to the macroelements we use the arithmetic average of the values at coarse grid nodes defining the macroelement. This situation can be seen in Fig. 4.
- Edge nodes:
 - If the fine grid node lies on a macroelement edge, then its value is defined to be the average of the 2 coarse grid nodes defining the macro-edge (in Fig. 4 such a node is j_1).
 - If the fine grid node lies on more than one macro-edge, then its value is defined to be the simple arithmetic average of all the values corresponding to the different edges (in Fig. 4 such a node is j_2).

As an example we give the interpolated values at fine grid nodes for the grids in Fig. 4):

$$\begin{aligned} I_{J-1}^J v_h(x_{j_1}) &= \frac{v_h(x_{k_1}) + v_h(x_{k_2})}{2}, \\ I_{J-1}^J v_h(x_{j_2}) &= \frac{v_h(x_{k_1}) + v_h(x_{k_2}) + v_h(x_{k_3})}{3}, \\ I_{J-1}^J v_h(x_j) &= \frac{v_h(x_{k_1}) + v_h(x_{k_2}) + v_h(x_{k_3}) + v_h(x_{k_4}) + v_h(x_{k_5})}{5}. \end{aligned}$$

This simple interpolation has the advantage that the matrix corresponding to it can be stored in the computer memory using only integers. The matrix vector multiplication is easier to perform and this basis preserves the constant function.

4.4. Multilevel implementation. A straightforward multilevel implementation of the coarsening algorithm, can be done by simply retriangulating the set of coarse grid points and apply the derefinement algorithm to the obtained triangulation. In this section we will propose another version, which has the advantage that it operates only on the graph and does not use nodal coordinates and real numbers arithmetic.

To apply the algorithm recursively, we need to define the same input data, but using the coarse grid. We first define the elements (triangles, or triples of vertices) on the coarse grid in the following way:

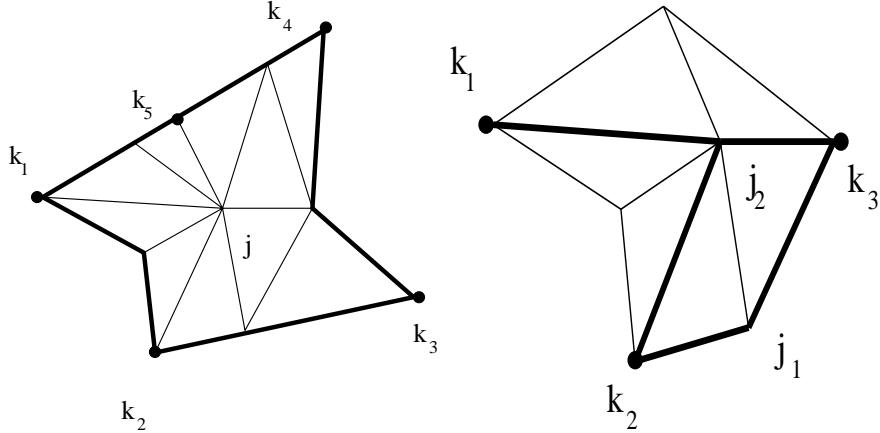


FIGURE 4. Example of interpolation. Thick lines mark the macroelement boundaries.

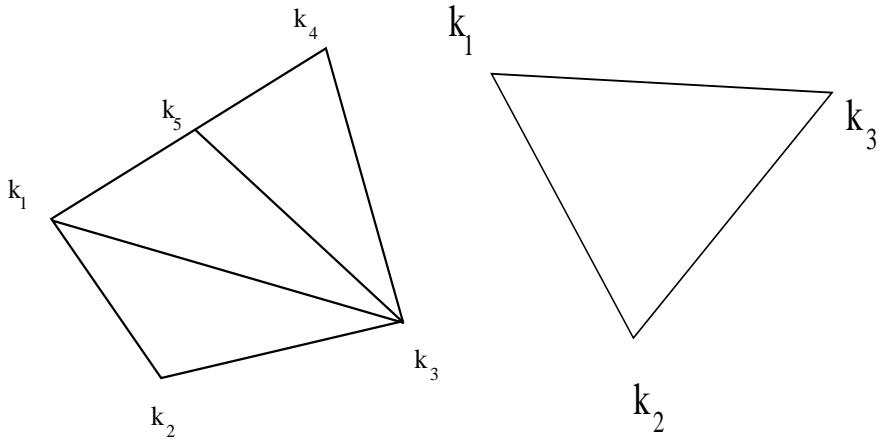


FIGURE 5. The coarse grid elements

- Consider every macroelement as a polygon with m vertices (k_1, k_2, \dots, k_m) in counter-clockwise ordering (m is the number of coarse grid vertices forming the macroelement). We triangulate it with $m-2$ triangles in the following way:
 1. If $m \leq 3$ stop.
 2. Form the triangles (k_1, k_2, k_3) and (k_1, k_3, k_m) .
 3. Remove k_1 and k_2 from the polygon, set $k_1 \leftarrow k_m$ and $k_{i-1} \leftarrow k_i$ for $i = 3, \dots, m-1$, $m \leftarrow m-2$. Go to 1.
- If a fine grid node lies on more than one macro-edge we form a m -gon with vertices the coarse grid points surrounding it (see Fig. 4, such a node is j_2). We triangulate this m -gon in the same way as we did in the previous step. Such a m -gon is shown in Fig. 5 on the right. This triangle corresponds to node j_2 in Fig. 4.

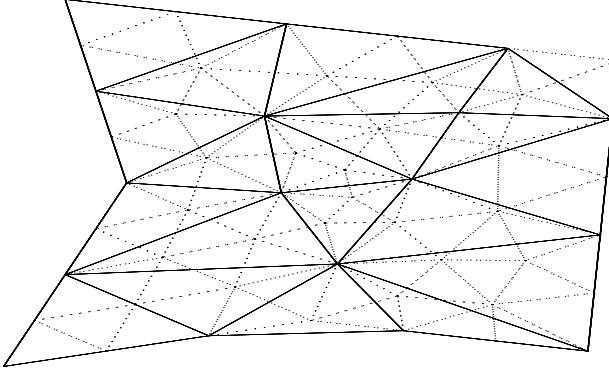


FIGURE 6. Coarse grid graph

The coarse grid configurations corresponding to Fig. 4 are given in Fig. 5. Thus we define the graph $G_c = (V_c, E_c)$ corresponding to the coarse grid to be the following:

- V_c : Vertices are the coarse grid nodes
- E_c : Two vertices are connected by an edge if and only if they are in one and the same triangle.
- V'_c : Vertices are the triangles we have formed.
- E'_c : Two triangles are connected by an edge if they share an edge in G_c .

The issue we have to address here is that in this way we might not get a valid finite element triangulation. It might happen that for some edge in G_c , there are more than 2 triangles adjacent to it. But as we pointed out before we only need an auxiliary graph G'_c and a correspondence between E_c and E'_c which we have defined. In Fig. 6 the graph G_c for the model finite element grid is plotted. As it can be seen, we obtained a valid triangulation and in practice this is often the case.

A simple application of the algorithm yields:

If the fine grid is obtained by a successive halving refinement and if the MIS on each level coincide the original coarse grid set, then the macro-elements will coincide exactly with the underlined (nested) coarse grids.

REMARK 5. Note that although the multilevel sequence of grids is non-nested the corresponding finite dimensional spaces are nested, because the basis in M_{k-1} is always defined as a linear combination of the fine grid basis via the interpolation. It is also clear from the definition that the arithmetic average interpolation preserves the constant function in each macroelement on all levels.

5. On the convergence

In this section, we briefly discuss the convergence of the aforementioned multigrid method. We shall prove a result concerning the stability and approximation properties of the agglomerated coarse spaces. As a result we can conclude that our multigrid algorithm converges uniformly if the number of levels is fixed. We are yet to extend our result to truly multilevel case.

Given a triangulation T_h and the corresponding linear finite element space $M_h \subset H^1(\Omega)$, let $M_H \subset M_h$ be obtained by the agglomeration algorithm described in the previous section. Let $Q_H : H^1(\Omega) \rightarrow M_H$ be the L^2 -projection. The assumption we make is for every macroelement G_H there exists an auxiliary big simplex K_H of diameter H , containing G_H together with all its neighboring elements from the fine grid. We also assume that $H/h \leq c$, for some constant c .

We claim that for every $v \in H^1(\Omega)$ the following stability and approximation properties hold:

$$(8) \quad \|Q_H v\|_{1,\Omega} \leq C\|v\|_{1,\Omega},$$

$$(9) \quad \|v - Q_H v\|_{0,\Omega} \leq CH|v|_{1,\Omega}.$$

We shall give detailed proof of our claim. Our proof is based on an averaged nodal value interpolation similar to the one described in Scott and Zhang [14]. Given any “coarse node” x_k , let F_k be an $n-1$ dimensional face from T_h that contains x_k . Let $\psi_k(x)$ be the linear function on F_k such that

$$\langle v, \psi_k \rangle_{0,F_k} = v(x_k) \quad \forall v \in \mathcal{P}_1(F_k).$$

Now define $\Pi_H : H^1(\Omega) \rightarrow M_H$ by

$$(\Pi_H v)(x_k) = \langle v, \psi_k \rangle_{0,F_k}$$

for each coarse node x_k , and the value of $\Pi_H v$ on all other fine grid nodes are determined by the prolongation operator. We claim that for any $v \in H^1(\Omega)$

$$(10) \quad \|\Pi_H v\|_{1,\Omega} \leq C\|v\|_{1,\Omega},$$

$$(11) \quad \|v - \Pi_H v\|_{0,\Omega} \leq CH|v|_{1,\Omega}.$$

We shall first prove (11). By the extension theorem, we may assume that $v \in H^1(\mathbb{R}^n)$ satisfying

$$|v|_{1,\mathbb{R}^n} \leq C|v|_{1,\Omega}.$$

Let now G_H be a macroelement. By construction we can find an auxiliary big simplex K_H (with diameter bounded by cH) that contains G_H together with all its neighboring elements from the fine grid. Now let us introduce the affine mapping $K_H \rightarrow \hat{K}$, where \hat{K} is the standard reference element. Correspondingly we will have $G_H \rightarrow \hat{G}$, $v \rightarrow \hat{v}$, and $\Pi_H v \rightarrow \hat{\Pi}\hat{v}$.

We now consider $\hat{\Pi}$. It is easy to see that by trace theorem we have

$$\|\hat{v} - \hat{\Pi}\hat{v}\|_{0,\hat{G}} \leq C\|\hat{v}\|_{1,\hat{K}}, \quad \forall \hat{v} \in H^1(\hat{K})$$

and by construction $\hat{\Pi}$ is invariant on constant functions, namely $\hat{\Pi}\hat{c} = \hat{c}$, for any $\hat{c} \in \mathbb{R}^1$. Therefore

$$\begin{aligned} \|\hat{v} - \hat{\Pi}\hat{v}\|_{0,\hat{G}} &= \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c} - \hat{\Pi}(\hat{v} + \hat{c})\|_{0,\hat{G}} \\ &\leq C \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c}\|_{1,\hat{K}} \leq C|\hat{v}|_{1,\hat{K}}. \end{aligned}$$

By scaling back to K_H we get

$$\|v - \Pi_H v\|_{0,G_H} \leq CH|v|_{1,K_H}.$$

Summing over all macroelements we have

$$\begin{aligned}\|v - \Pi_H v\|_{0,\Omega}^2 &\leq \sum_{G_H \subset \Omega} \|v - \Pi_H v\|_{0,G_H}^2 \\ &\leq CH^2 \sum_{K_H \supset G_H} |v|_{1,K_H}^2 \\ &\leq CH^2 |v|_{1,\mathbb{R}^n}^2 \leq CH^2 |v|_{1,\Omega}^2.\end{aligned}$$

This proves (11)

We shall now prove (10). The proof uses the standard scaling argument and invariance of $\hat{\Pi}$ on $\mathcal{P}_0(\hat{K})$. We have

$$\begin{aligned}|\Pi_H v|_{1,G_H} &\leq CH^{\frac{n}{2}-1} |\hat{\Pi} \hat{v}|_{1,\hat{G}} = CH^{\frac{n}{2}-1} \inf_{\hat{c} \in \mathbb{R}^1} |\hat{\Pi}(\hat{v} + \hat{c})|_{1,\hat{G}} \\ &\leq CH^{\frac{n}{2}-1} \inf_{\hat{c} \in \mathbb{R}^1} \|\hat{v} + \hat{c}\|_{1,\hat{K}} \leq CH^{\frac{n}{2}-1} |\hat{v}|_{1,\hat{K}}.\end{aligned}$$

By scaling back to K_H we get the desired estimate (10).

Consequently

$$\|v - Q_H v\|_{0,\Omega} \leq \|v - \Pi_H v\|_{0,\Omega} \leq CH |v|_{1,\Omega}.$$

and

$$\begin{aligned}|Q_H v|_{1,\Omega} &\leq |Q_H v - \Pi_H v|_{1,\Omega} + |\Pi_H v|_{1,\Omega} \\ &\leq C(h^{-1} \|Q_H v - \Pi_H v\|_{0,\Omega} + \|v\|_{1,\Omega}) \leq C |v|_{1,\Omega}.\end{aligned}$$

By the convergence theory in Bramble, Pasciak, Wang, Xu [4] we use the estimates (8) and (9) to conclude that: *the agglomeration multigrid algorithm converges uniformly with respect to h if the number of levels is fixed.*

6. Numerical examples

We consider the Laplace equation:

$$(12) \quad \begin{cases} -\Delta u = 1, & (x, y) \in \Omega \subset \mathbb{R}^2, \\ u(x, y) = 0, & (x, y) \in \partial\Omega. \end{cases}$$

In these examples we use the standard V -cycle preconditioner and the outer acceleration is done by the conjugate gradient method. In the V -cycle we use 1-pre and 1-post smoothing steps. The smoothing operator is forward Gauß-Seidel. The PCG iterations are terminated when the relative residual is less than 10^{-6} . We also present the examples using the variable V -cycle, doubling the smoothing steps on each level. We are interested in checking numerically the convergence of PCG preconditioned with V -cycle based on the simple interpolation we derived.

In Figures 7–8, we plot the macroelements for different unstructured grids and different number of levels to illustrate the coarsening algorithm. These fine grids are obtained by Delaunay triangulation of randomly placed point sets. They are not obtained by any refinement procedure. Figure 9 shows the convergence histories for a varying number of unknowns on two types of grids. One of these (one-element airfoil) has one internal boundary, the other one has four internal boundaries.

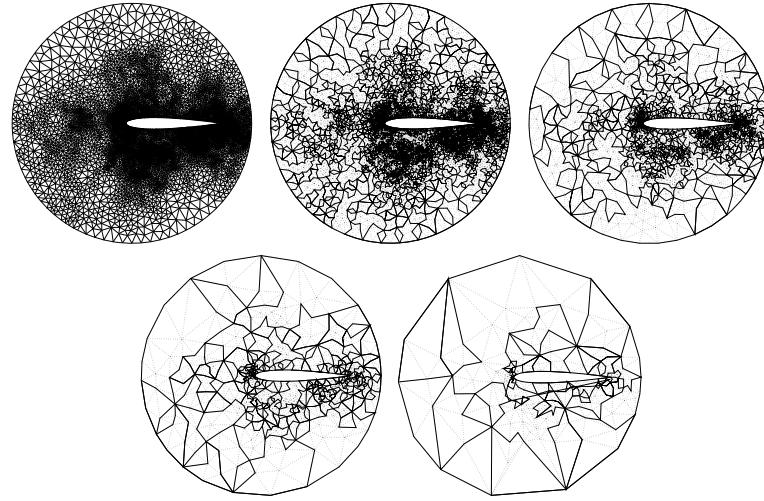


FIGURE 7. Macroelements for one element airfoil: level= 5 $N_h = 12665$; level= 4 $N_H^1 = 3404$; level= 3 $N_H^2 = 928$; level= 2 $N_H^3 = 257$; level= 1 $N_H^4 = 74$; level= 0 $N_H^5 = 24$.

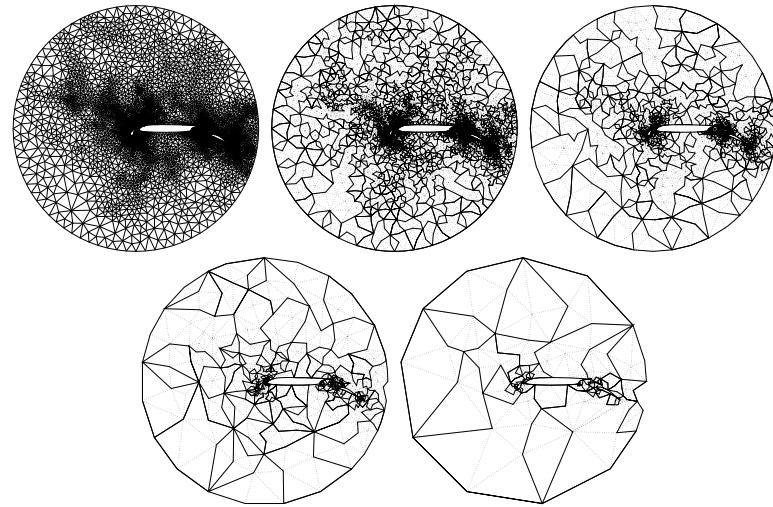


FIGURE 8. Macroelements for four element airfoil: level= 5 $N_h = 12850$; level= 4 $N_H^1 = 3444$; level= 3 $N_H^2 = 949$; level= 2 $N_H^3 = 270$; level= 1 $N_H^4 = 80$; level= 0 $N_H^5 = 26$.

As interpolation, we use the one described in Section 4.3. The numerical experiments suggest that for isotropic problems (such as Laplace equation), the convergence of the variable V-cycle seems to be uniform with respect to the mesh size h .

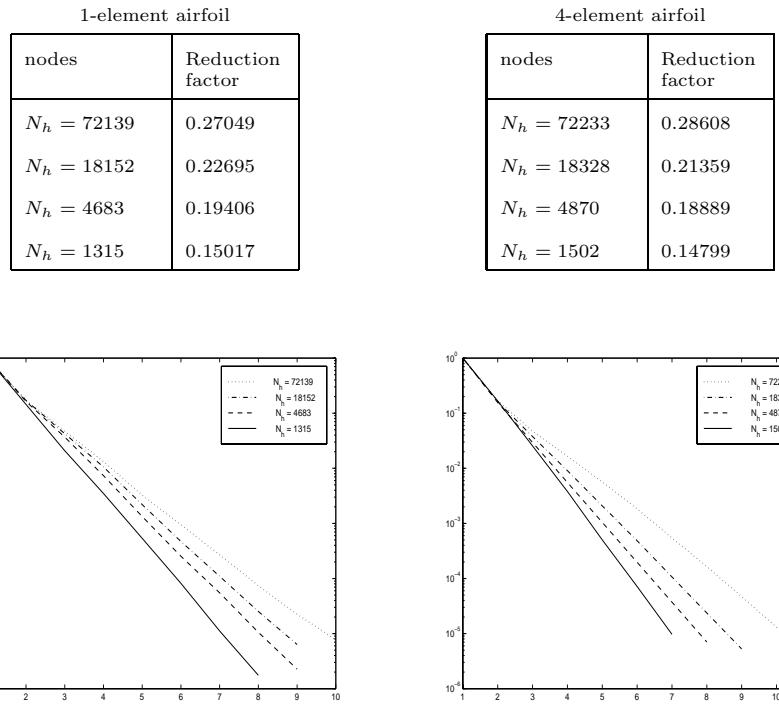


FIGURE 9. Convergence history and average reduction per iteration for varying number of unknowns, V -cycle

References

1. R. Bank and J. Xu, *An algorithm for coarsening unstructured meshes*, Numer. Math. **73** (1996), no. 1, 1–36.
2. D. Braess, *Towards algebraic multigrid for elliptic problems of second order*, Computing **55** (1995), 379–393.
3. J. Bramble, *Multigrid methods*, Pitman, Notes on Mathematics, 1994.
4. James H. Bramble, Joseph E. Pasciak, Junping Wang, and Jinchao Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp. **57** (1991), no. 195, 23–45.
5. T. F. Chan and Barry Smith, *Domain decomposition and multigrid methods for elliptic problems on unstructured meshes*, Domain Decomposition Methods in Science and Engineering, Proceedings of the Seventh International Conference on Domain Decomposition, October 27–30, 1993, The Pennsylvania State University (David Keyes and Jinchao Xu, eds.), American Mathematical Society, Providence, 1994, also in Electronic Transactions on Numerical Analysis, v.2, (1994), pp. 171–182.
6. P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multi-grid by smoothed aggregation for second and forth order elliptic problems*, Computing **56** (1996), 179–196.
7. P. Vaněk and J. Křížková, *Two-level preconditioner with small coarse grid appropriate for unstructured meshes*, Numer. Linear Algebra Appl. **3** (1996), no. 4, 255–274.
8. H. Guillard, *Node-nested multi-grid method with Delaunay coarsening*, Tech. Report RR-1898, INRIA, Sophia Antipolis, France, March 1993.
9. W. Hackbusch, *Multi-grid methods and applications*, Springer Verlag, New York, 1985.
10. W. Hackbusch and S. A. Sauter, *Composite finite elements for the approximation of PDEs on domains with complicated micro-structures*, Numerische Mathematik **75** (1995), 447–472.

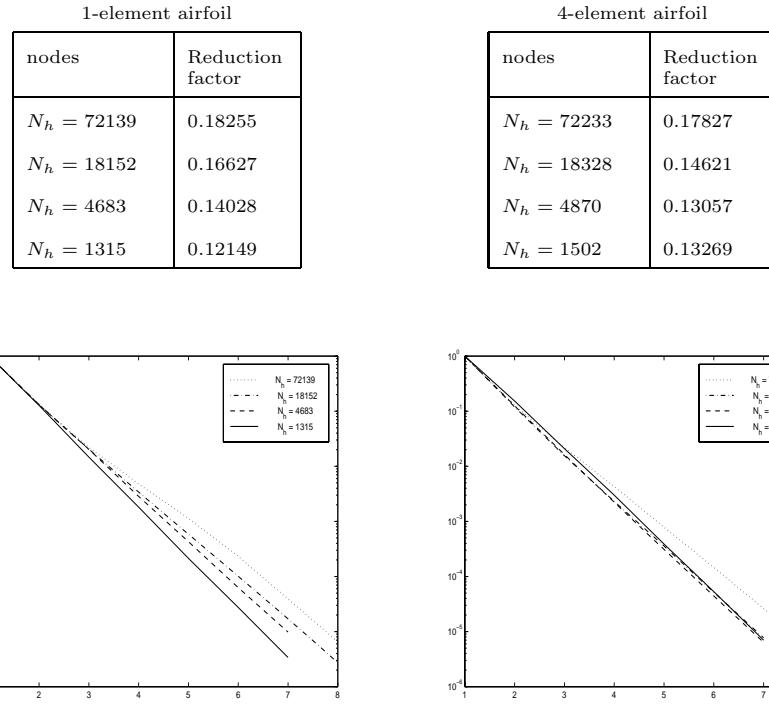


FIGURE 10. Convergence history and average reduction per iteration for varying number of unknowns, variable V -cycle

11. B. Koobus, M. H. Lallemand, and A. Dervieux, *Unstructured volume-agglomeration MG: solution of the Poisson equation*, International Journal for Numerical Methods in Fluids **18** (1994), no. 1, 27–42.
12. A. A. Reusken, *A multigrid method based on incomplete Gaussian elimination*, Numer. Linear Algebra Appl. **3** (1996), no. 5, 369–390.
13. J. W. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid methods (Philadelphia, Pennsylvania) (S. F. McCormick, ed.), Frontiers in applied mathematics, SIAM, 1987, pp. 73–130.
14. L. R. Scott and S. Zhang, *Finite element interpolation of nonsmooth functions satisfying boundary conditions*, Math. Comp. **54** (1990), 483–493.
15. J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review **34** (1992), 581–613.
16. J. Xu, *The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids*, Computing **56** (1996), 215–235.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA AT LOS ANGELES, 405 HILGARD AVE, LOS ANGELES, CA 90024

E-mail address: chan@math.ucla.edu

CENTER FOR COMPUTATIONAL MATHEMATICS AND APPLICATIONS, DEPARTMENT OF MATHEMATICS, PENNSYLVANIA STATE UNIVERSITY, STATE COLLEGE, PA-16801

E-mail address: xu@math.psu.edu

CENTER FOR COMPUTATIONAL MATHEMATICS AND APPLICATIONS, DEPARTMENT OF MATHEMATICS, PENNSYLVANIA STATE UNIVERSITY, STATE COLLEGE, PA-16801

E-mail address: litz@math.psu.edu

Solution of Coercive and Semicoercive Contact Problems by FETI Domain Decomposition

Zdeněk Dostál, Ana Friedlander, and Sandra A. Santos

1. Introduction

A new Neumann-Neumann type domain decomposition algorithm for the solution of contact problems of elasticity and similar problems is described. The discretized variational inequality that models the equilibrium of a system of elastic bodies in contact is first turned by duality to a strictly convex quadratic programming problem with either box constraints or box and equality constraints. This step may be considered a variant of the FETI domain decomposition method where the subdomains are identified with the bodies of the system. The resulting quadratic programming problem is then solved by algorithms proposed recently by the authors. Important new features of these algorithms are efficient adaptive precision control on the solution of the auxiliary problems and effective application of projections, so that the identification of a priori unknown contact interfaces is very fast.

We start our exposition by reviewing a variational inequality in displacements that describes the conditions of equilibrium of a system of elastic bodies in contact without friction. The inequality enhances the natural decomposition of the spatial domain of the problem into subdomains that correspond to the bodies of the system, and we also indicate how to refine this decomposition. After discretization, we get a possibly indefinite quadratic programming problem with a block diagonal matrix.

A brief inspection of the discrete problem shows that its structure is not suitable for computations. The main drawbacks are the presence of general constraints that prevent effective application of projections, and a semidefinite or ill conditioned matrix of the quadratic form that may cause extremely expensive solutions of the auxiliary problems.

A key observation is that both difficulties may be essentially reduced by the application of duality theory. The matrix of the dual quadratic form turns out to be regular, moreover its spectrum is much more favorably distributed for application of the conjugate gradient based methods than the spectrum of the matrix of the

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N30, 35J85, 90C20, 90C90.

This research has been supported by CNPq, FAPESP 95/6574-9 and by grants GAČR 201/97/0421, 105/95/1273.

quadratic form arising from the discretization. These conclusions follow from the close relation of the procedure to the FETI method proposed by Farhat and Roux [20, 18] for the solution of linear elliptic problems. Furthermore, the inequality constraints of the dual problem are just non-negativity constraints, so that our recent results on application of projections and adaptive precision control may be used for the solution of these problems.

The structure of the constraints of the dual problem depends on the coercivity of the contact problem under consideration. If the contact problem is *coercive*, i.e. if prescribed equality constraints on the displacement of each body prevent its rigid body motion, then the dual problem has only simple non-negativity constraints. We describe an efficient algorithm for the solution of these problems that uses the conjugate gradient method with projections and inexact solution of the auxiliary subproblems that has been proposed independently by Friedlander and Martínez [3, 21, 22, 23, 24, 25, 26] and Dostál [11]. The algorithm has been proved to converge to the solution and conditions that guarantee the finite termination property have been established. The algorithm may be implemented with projections so that it can drop or add many couples of nodes on the contact interface whenever the active set is changed. Thus the contact interface may be identified very fast even with a poor initial guess.

Next we consider the solution of *semicoercive* problems, i.e. problems with ‘floating’ bodies. Application of duality reduces these problems to the solution of quadratic programming problems with simple bounds and equality constraints. In this case, the feasible set is too complex to enable effective evaluations of projections, but we use a variant of the augmented Lagrangian algorithm proposed for the solution of more general non-linear problems by Conn, Gould and Toint [5, 6]. The algorithm generates in the outer loop the Lagrange multipliers for equality constraints while auxiliary problems with simple inequality constraints are solved in the inner loop. The precision of the solution of the auxiliary problems is controlled by the norm of the violation of the equality constraints, and an estimate for the error has been obtained that does not have any term that accounts for the precision of the solution of the auxiliary problems with simple bounds. Results on global convergence and boundedness of the penalty parameter are also reported. Moreover, we show that the penalty term in the augmented Lagrangians affects the convergence of the conjugate gradient solution of the auxiliary problems only very mildly. The paper is completed by numerical experiments.

To simplify our exposition, we have restricted our attention to the frictionless contact problems. However, the algorithm may be extended to the solution of contact problems with Coulomb friction [16].

2. Conditions of equilibrium of elastic bodies

Consider a system of s homogeneous isotropic elastic bodies, each of which occupies in a reference configuration a domain Ω^p in \mathbb{R}^d , $d = 2, 3$ with sufficiently smooth boundary Γ^p as in Figure 1. We assume that the bodies do not interpenetrate each other so that the intersection of any two different domains is empty. Suppose that each Γ^p consists of three disjoint parts Γ_U^p , Γ_F^p and Γ_C^p , $\Gamma^p = \Gamma_U^p \cup \Gamma_F^p \cup \Gamma_C^p$, and that the displacements $\mathbf{U}^p : \Gamma_U^p \rightarrow \mathbb{R}^d$ and forces $\mathbf{F}^p : \Gamma_F^p \rightarrow \mathbb{R}^d$ are given. The part Γ_C^p denotes the part of Γ^p that may get into contact with some other body. In particular, we shall denote by Γ_C^{pq} the part of Γ^p that can be, in the solution, in

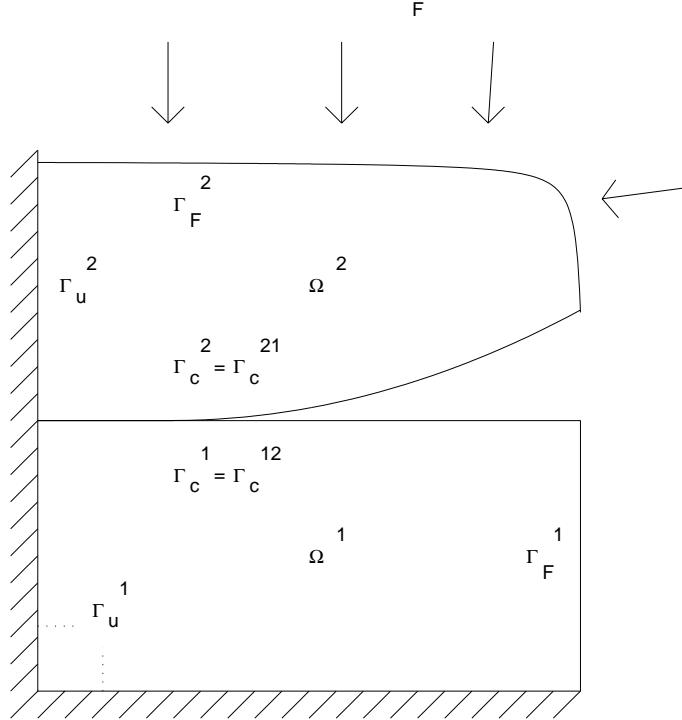


FIGURE 1. Contact problem

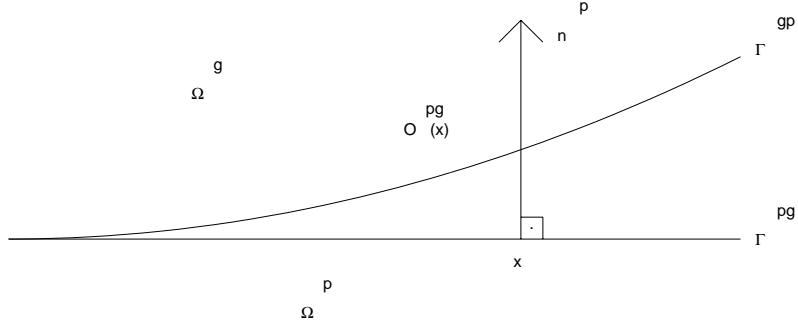


FIGURE 2. Linearized non-interpenetration

contact with the body Ω^q . Finally, let $c_{ijkl}^p : \Omega^p \rightarrow \mathbb{R}^d$ and $\mathbf{g}^p : \Omega^p \rightarrow \mathbb{R}^d$ denote the entries of the elasticity tensor and a vector of body forces, respectively.

For any sufficiently smooth displacements $\mathbf{u} : \Omega^1 \times \dots \times \Omega^s \rightarrow \mathbb{R}^d$, the total potential energy is defined by

$$(1) \quad J(\mathbf{u}) = \sum_{p=1}^s \left\{ \frac{1}{2} \int_{\Omega^p} a(\mathbf{u}^p, \mathbf{u}^p) d\Omega - \int_{\Omega^p} (\mathbf{g}^p)^T \mathbf{u}^p d\Omega - \int_{\Gamma_F^p} (\mathbf{F}^p)^T \mathbf{u}^p d\Gamma \right\}$$

where

$$(2) \quad a^p(\mathbf{u}^p, \mathbf{v}^p) = \frac{1}{2} \int_{\Omega^p} c_{ijk\ell} e_{ij}^p(\mathbf{u}^p) e_{k\ell}^p(\mathbf{v}^p) d\Gamma$$

$$(3) \quad e_{k\ell}^p(\mathbf{u}^p) = \frac{1}{2} \left(\frac{\partial u_k^p}{\partial x_\ell^p} + \frac{\partial u_\ell^p}{\partial x_k^p} \right).$$

We suppose that the elasticity tensor satisfies natural physical restrictions so that

$$(4) \quad a^p(\mathbf{u}^p, \mathbf{v}^p) = a(\mathbf{v}^p, \mathbf{u}^p) \text{ and } a(\mathbf{u}^p, \mathbf{u}^p) \geq 0.$$

To describe the linearized non-interpenetration conditions, let us define for each $p < q$ a one-to-one continuous mapping $\mathbf{O}^{pq} : \Gamma_C^{pq} \rightarrow \Gamma_C^{qp}$ that assigns to each $\mathbf{x} \in \Gamma_C^{pq}$ some point of Γ_C^{qp} that is near to \mathbf{x} as in Figure 2. The linearized non-interpenetration condition at $\mathbf{x} \in \Gamma_C^{pq}$ then reads

$$(5) \quad (\mathbf{u}^p(\mathbf{x}) - \mathbf{u}^q(\mathbf{O}^{pq}(\mathbf{x}))) \mathbf{n}^p \leq (\mathbf{O}^{pq}(\mathbf{x}) - \mathbf{x}) \mathbf{n}^p, \mathbf{x} \in \Gamma_C^{pq}, p < q.$$

Similar conditions may be written for description of non-interpenetration with rigid support.

Now let us introduce the Sobolev space

$$(6) \quad \mathcal{V} = H^1(\Omega^1)^d \times \dots \times H^1(\Omega^s)^d,$$

and let $\mathbf{K} = \mathbf{K}_{eq} \cap \mathbf{K}_{ineq}$ denote the set of all kinematically admissible displacements, where

$$(7) \quad \mathbf{K}_{eq} = \{ \mathbf{v} \in \mathcal{V} : \mathbf{v}^p = \mathbf{U} \text{ on } \Gamma_U^p \}$$

and

$$(8) \quad \mathbf{K}_{ineq} = \{ \mathbf{v} \in \mathcal{V} : (\mathbf{v}^p(\mathbf{x}) - \mathbf{v}^q(\mathbf{O}^{pq}(\mathbf{x}))) \mathbf{n}^p \leq (\mathbf{O}^{pq}(\mathbf{x}) - \mathbf{x}) \mathbf{n}^p, \mathbf{x} \in \Gamma_C^{pq}, p < q \}.$$

The displacement $\mathbf{u} \in \mathbf{K}$ of the system of bodies in equilibrium satisfies

$$(9) \quad J(\mathbf{u}) \leq J(\mathbf{v}) \text{ for any } \mathbf{v} \in \mathbf{K}.$$

Conditions that guarantee the existence and uniqueness may be found e.g. in [4, 28].

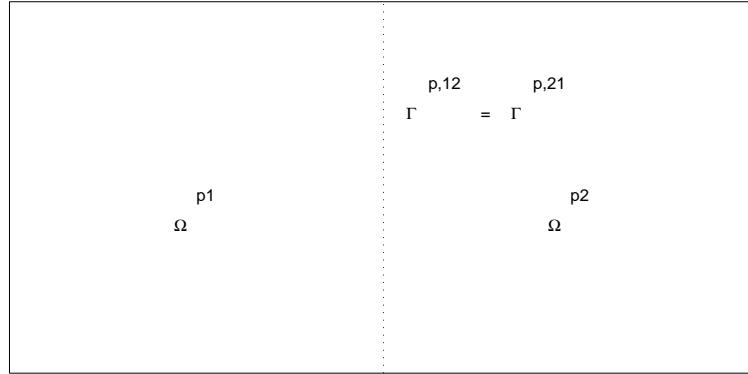
More general boundary conditions than those described by prescribed forces or displacements may be considered, e.g. prescribed normal displacements and zero forces in the tangential plane. Moreover, we can also decompose each body into subdomains as in Figure 3 to obtain optional secondary decomposition. The Sobolev space \mathcal{V} would then be defined on the product of all subdomains and the definition of the set \mathbf{K} would enhance also the interface equality constraints that guarantee continuity of the displacements across new auxiliary interfaces $\Gamma_A^{p,ij}$ in each block Ω^p .

3. Discretized contact problem on interface

If there is no secondary decomposition, then the finite element discretization of $\Omega = \Omega^1 \cup \dots \cup \Omega^s$ with suitable numbering of the nodes results in the quadratic programming (QP) problem

$$(10) \quad \frac{1}{2} \mathbf{u}^T K \mathbf{u} - \mathbf{f}^T \mathbf{u} \rightarrow \min \text{ subject to } B \mathbf{u} \leq \mathbf{c}$$

with a symmetric positive definite or positive semidefinite block-diagonal matrix $K = \text{diag}(K_1, \dots, K_s)$ of order n , an $m \times n$ full rank matrix B , $\mathbf{f} \in \mathbb{R}^n$, and

FIGURE 3. Secondary decomposition of Ω^p

$c \in I\!\!R^m$. The matrix B and the vector c describe the linearized incremental non-interpenetration conditions. The rows b_i of B are formed by zeros and appropriately placed coordinates of outer unit normals, so that the change of normal distance due to the displacement u is given by $u^T b_i$, and the entry c_i of c describes the normal distance of the i -th couple of corresponding nodes on the contact interface in the reference configuration. The vector f describes the nodal forces arising from the volume forces and/or some other imposed tractions. Typically n is large and m is much smaller than n . The diagonal blocks K_p that correspond to subdomains Ω^p are positive definite or semidefinite sparse matrices. Moreover, we shall assume that the nodes of the discretization are numbered in such a way that the matrices K_i are banded matrices that can be effectively decomposed, possibly after some regularization, by means of the Cholesky factorization.

If there is a secondary decomposition, then the continuity of the displacements across the auxiliary interface requires $u^T h_i = 0$, where h_i are vectors of order n with zero entries except 1 and -1 in appropriate positions. If H is a matrix formed by the rows h_i , then the discretization of problem (10) with the secondary decomposition results in the QP problem

$$(11) \quad \frac{1}{2} u^T K u - f^T u \rightarrow \min \text{ subject to } Bu \leq c \text{ and } Hu = 0.$$

With a suitable enumeration of the nodes, each K_i turns out to be block diagonal with banded diagonal blocks.

Even though (11) is a standard convex quadratic programming problem, its formulation is not suitable for numerical solution. The reasons are that K might be singular and the feasible set is in general so complex that projections cannot be computed to obtain fast identification of the active set at the solution.

The complications mentioned above may be essentially reduced by applying the duality theory of convex programming (e.g. Dostál [10, 9]). If there is no secondary decomposition, we get

$$(12) \quad \theta(\lambda) \rightarrow \min \text{ subject to } \lambda \geq 0 \text{ and } R^T(f - B^T \lambda) = 0$$

where

$$(13) \quad \theta(\lambda) = \frac{1}{2} \lambda^T B K^+ B^T \lambda - \lambda^T (B K^+ f - c),$$

R denotes a matrix whose columns span the null space of K , and K^+ denotes a generalized inverse of K that satisfies $KK^+K = K$. Let us recall that

$$K^+ = \text{diag}(K_1^+, \dots, K_p^+)$$

and that $K_p^+ = K_p^{-1}$ whenever K_p is non-singular. If K_p is singular then it is easy to check that there is a permutation matrix P_p and a non-singular matrix F_p such that

$$(14) \quad P_p^T K_p P_p = \begin{pmatrix} F_p & S_p \\ S_p^T & S_p^T F_p^{-1} \end{pmatrix}$$

and

$$(15) \quad K_p^+ = P_p \begin{pmatrix} F_p^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_p^T.$$

Once the solution λ of (12) is known, the vector u that solves (10) can be evaluated. In particular, if K is positive definite then

$$(16) \quad u = K^{-1}(f - B^T \lambda).$$

If K is singular and R is a full rank matrix then

$$(17) \quad u = R\alpha + K^+(f - B^T \lambda),$$

with

$$(18) \quad \alpha = (R^T \tilde{B}^T \tilde{B} R)^{-1} R^T \tilde{B}^T (\tilde{c} - \tilde{B} A^+(f - B^T \lambda))$$

and (\tilde{B}, \tilde{c}) formed by the rows of (B, c) that correspond to the nonzero entries of λ .

If there is a secondary decomposition, then there are additional Lagrange multipliers for equalities. Thus, the only new feature when compared with the problem without the secondary decomposition is the presence of free Lagrange multipliers in the dual formulation in this case.

The matrix BK^+B^T is invertible when no rigid body displacement can be written as a linear combination of the columns of B^T . Moreover, the matrix BK^+B^T is closely related to the matrix resulting from the application of the FETI method of Farhat and Roux [20], so that its spectrum is relatively favorably distributed for the application of the conjugate gradient method (Farhat, Mandel and Roux [17, 19]).

4. Solution of coercive problems

An important point in the development of an efficient algorithm for the solution of (12) is the observation that QP problems with simple bounds are much simpler than more general QP problems. Here we shall briefly review our results on the solution of QP problems with simple bounds.

To simplify our notations, let us denote

$$\begin{array}{lll} A & = & BK^+B^T & b & = & BK^+f - c \\ d & = & R^T c & D & = & R^T B^T \end{array}$$

and let us first assume that K is non-singular, so that problem (12)-(13) reads

$$(19) \quad \theta(x) \rightarrow \min \quad \text{subject to} \quad x \geq 0$$

where

$$(20) \quad \theta(x) = \frac{1}{2}x^T Ax - b^T x.$$

Let us denote by $\mathcal{A}(x)$ and $\mathcal{F}(x)$ the active and free sets of indices of x , respectively, i.e.

$$(21) \quad \mathcal{A}(x) = \{i : x_i = 0\} \quad \text{and} \quad \mathcal{F}(x) = \{i : x_i \neq 0\}.$$

The unbalanced contact gradient g^C and the inner gradient g^I of $\theta(x)$ are defined by

$$(22) \quad g_i^I = g_i \text{ for } i \in \mathcal{F}(x) \text{ and } g_i^I = 0 \text{ for } i \in \mathcal{A}(x)$$

$$(23) \quad g_i^C = 0 \text{ for } i \in \mathcal{F}(x) \text{ and } g_i^C = g_i^- \text{ for } i \in \mathcal{A}(x)$$

where $g = g(x) = \nabla \theta(x)$, $g_i = g_i(x)$ and $g_i^- = \min\{0, g_i\}$. Hence the Kuhn-Tucker conditions for the solution of (19) are satisfied when the projected gradient $g^P = g^I + g^C$ vanishes.

An efficient algorithm for the solution of convex QP problems with simple bounds has been proposed independently by Friedlander and Martínez [21] and Dostál [11]. The algorithm may be considered a modification of the Polyak algorithm that controls the precision of the solution of auxiliary problems by the norm of g^C in each inner iterate y^i .

If for $\Gamma > 0$ the inequality

$$\|g^C(y^i)\| \leq \Gamma \|g^I(y^i)\|$$

holds then we call y^i proportional. The algorithm explores the face

$$W_I = \{y : y_i = 0 \text{ for } i \in I\}$$

with a given active set I as long as the iterates are proportional. If y^i is not proportional, we generate y^{i+1} by means of the descent direction $d^i = -g^C(y^i)$ in a step that we call proportioning, and then we continue exploring the new face defined by $I = \mathcal{A}(y^{i+1})$. The class of algorithms driven by proportioning may be defined as follows.

ALGORITHM 1. (General Proportioning Scheme - GPS)

Let $y^0 \geq 0$ and $\Gamma > 0$ be given. For $i > 0$, choose y^{i+1} by the following rules:

- (i) If y^i is not proportional, define y^{i+1} by proportioning.
- (ii) If y^i is proportional, choose $y^{i+1} \geq 0$ so that

$$\theta(y^{i+1}) \leq \theta(y^i)$$

and y^{i+1} satisfies at least one of the conditions: $\mathcal{A}(y^i) \subset \mathcal{A}(y^{i+1})$, y^{i+1} is not proportional, or y^{i+1} minimizes $\theta(\xi)$ subject to $\xi \in W_I$, $I = \mathcal{A}(y^i)$.

The set relation \subset is used in the strict sense so that it is satisfied if the set on the left is a proper subset of the set on the right. Basic theoretical results have been proved in [3, 11, 21, 22].

THEOREM 2. Let x^k denote an infinite sequence generated by Algorithm GPS with given x^0 and $\Gamma > 0$. Let $\theta(x)$ be a strictly convex quadratic function. Then the following statements are true:

- (i) x^k converges to the solution \bar{x} of (19).
- (ii) If the problem (19) is not degenerate, then there is k such that $\bar{x} = x^k$.
- (iii) If $\Gamma \geq \kappa(A)^{1/2}$, where $\kappa(A)$ is the spectral condition number of A , then there is k such that $\bar{x} = x^k$.

Step (ii) of Algorithm GPS may be implemented by means of the conjugate gradient method. The most simple implementation of this step starts from $y^0 = x^k$ and generates the conjugate gradient iterations y^1, y^2, \dots for $\min\{\theta(y) : y \in \mathcal{W}_I, I = \mathcal{A}(y^0)\}$ until y^i is found that is not feasible or not proportional or minimizes $\theta(x)$ subject to $y \geq 0$. If y^i is feasible, then we put $x^{k+1} = y^i$, otherwise $y^i = y^{i-1} - \alpha^i p^i$ is not feasible and we can find $\tilde{\alpha}^i$ so that $x^{k+1} = y^i - \tilde{\alpha}^i p^i$ is feasible and $\mathcal{A}(x^k) \not\subseteq \mathcal{A}(x^{k+1})$. We shall call the resulting algorithm *feasible proportioning* [11].

An obvious drawback of feasible proportioning is that the algorithm is usually unable to add more than one index to the active set in one iteration. A simple but efficient alternative is to replace the feasibility condition by $\theta(Py^{i+1}) \leq \theta(Py^i)$, where Py denotes the projection on the set $\Omega = \{y : y \geq 0\}$. If the conjugate gradient iterations are interrupted when condition $\theta(Py^{i+1}) > \theta(Py^i)$ is satisfied, then a new iteration is defined by $x^{k+1} = Py^i$. Resulting modification of the feasible proportioning algorithm is called *monotone proportioning* [11]. More details on implementation of the algorithm may be found in [15].

5. Solution of semicoercive problems

Now we shall assume that the matrix K is only positive semidefinite, so that problem (12)-(13) with the notations of the previous section reads

$$(24) \quad \theta(x) \rightarrow \min \quad \text{subject to} \quad x \geq 0 \text{ and } Dx = d.$$

The algorithm that we propose here may be considered a variant of the algorithm proposed by Conn, Gould and Toint(1991) for identification of stationary points of more general problems.

ALGORITHM 3. (*Simple bound and equality constraints*)

Step 0. { Initialization of parameters } Set $0 < \alpha < 1$ [$\alpha = .1$] for equality precision update, $1 < \beta$ [$\beta = 100$] for penalty update, $\rho_0 > 0$ [$\rho_0 = 100$] for initial penalty parameter, $\eta_0 > 0$ [$\eta_0 = .001$] for initial equality precision, $M > 0$ [$M = \rho_0/100$] for balancing ratio, μ^0 [$\mu^0 = 0$] and $k = 0$.

Step 1. Find x^k so that

$$\|g^P(x^k, \mu^k, \rho_k)\| \leq M\|Dx^k - d\|.$$

Step 2. If $\|g^P(x^k, \mu^k, \rho_k)\|$ and $\|Dx^k - d\|$ are sufficiently small then x^k is the solution.

Step 3. $\mu^{k+1} = \mu^k + \rho_k(Dx^k - d)$.

Step 4. If $\|Dx^k - d\| \leq \eta_k$ 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 and return to Step 1.

In this algorithm, we now denote by g the gradient of the augmented Lagrangian

$$L(x, \mu, \rho) = \theta(x) + \mu^T Dx + \frac{1}{2}\rho\|Dx - d\|^2$$

so that

$$g(x, \mu, \rho) = Ax - b + D^T(\mu + \rho D(x - d))$$

An implementation of Step 1 is carried out by minimization of the augmented Lagrangian L subject to $x \geq 0$ by means of the algorithm of the previous section. The unique solution $\bar{x} = \bar{x}(\mu, \rho)$ of this auxiliary problem satisfies the Kuhn-Tucker conditions

$$(25) \quad g^P(\bar{x}, \mu, \rho) = 0.$$

Typical values of the parameters are given in brackets.

The essential feature of this algorithm is that it deals completely separately with each type of constraint and that it accepts inexact solutions of the auxiliary box constrained problems in Step 1. For parallel implementation, A should be kept as the product BK^+B since A is just used in the matrix-vector products. The action of K^+ may be evaluated by means of a triangular decomposition.

The algorithm has been proved ([13]) 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 exact solution of auxiliary quadratic programming problems (i.e. $M = 0$) and the penalty parameter is uniformly bounded.

The use of the augmented Lagrangian method turned out to be very efficient in spite of the fact that it obviously reintroduces ill conditioning into the auxiliary problems. The explanation is given by the following theorem and by analysis of the conjugate gradient method by Axelsson and Lindskog [1, 2], who showed that the rate of convergence is much faster than it could be expected from the conditioning of the problem provided there is a gap in the spectrum.

THEOREM 4. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix, $D \in \mathbb{R}^{m \times n}$ a full rank matrix, $m < n$ and $\rho > 0$. For any matrix M of order n , let $\delta_1(M) \leq \dots \leq \delta_n(M)$ denote the eigenvalues of M . Then*

$$(26) \quad \delta_{n-m}(A + \rho D^T D) \leq \delta_n(A)$$

$$(27) \quad \delta_{n-m+1}(A + \rho D^T D) \geq \rho \delta_{n-m+1}(D^T D) > 0.$$

6. Numerical experiments

In this section, we illustrate the practical behavior of our algorithm. First, a model problem used to validate the algorithm is presented. Next, two problems arising in mechanical and mining engineering, respectively, are commented. All the experiments were run in a PC-486 type computer, DOS operating system, Microsoft Fortran 77 and double precision. The auxiliary problems were solved by QUACAN routine developed in the Institute of Mathematics, Statistics and Scientific Computation of UNICAMP.

PROBLEM 1. This is a model problem resulting from the finite difference discretization of the following continuous problem:

$$\begin{aligned} \text{Minimize } q(u_1, u_2) &= \sum_{i=1}^2 \left(\int_{\Omega_i} |\nabla u_i|^2 d\Omega - \int_{\Omega_i} P u_i d\Omega \right) \\ \text{subject to } u_1(0, y) &\equiv 0 \text{ and } u_1(1, y) \leq u_2(1, y) \text{ for } y \in [0, 1], \end{aligned}$$

where $\Omega_1 = (0, 1) \times (0, 1)$, $\Omega_2 = (1, 2) \times (0, 1)$, $P(x, y) = -1$ for $(x, y) \in (0, 1) \times [0.75, 1]$, $P(x, y) = 0$ for $(x, y) \in (0, 1) \times (0, 0.75)$, $P(x, y) = -1$ for $(x, y) \in (1, 2) \times (0, 0.25)$ and $P(x, y) = 0$ for $(x, y) \in (1, 2) \times (0.25, 1)$. The discretization scheme consists in a regular grid of 21×21 nodes for each unitary interval. We took the identically zero initial approximation. This problem is such that the matrix of the quadratic function is singular due to the lack of Dirichlet data on the boundary of Ω_2 . In order to reduce the residual to 10^{-5} , three simple bounded (SB) problems had to be solved. The total number of iteration used by QUACAN was 23, taking 34 matrix-vector products. More details on this problem may be found in [10].

PROBLEM 2. The objective of this problem is to identify the contact interface and evaluate the contact stresses of a system of elastic bodies in contact. Some rigid motion is admitted for these bodies. This type of problems is treated in [14]. The model problem considered to test our algorithm consists of two identical cylinders that lie one above the other on a rigid support. A vertical traction is applied at the top 1/12 of the circumference of the upper cylinder. Assuming the plane stress, the problem was reduced to 2D and discretized by the boundary element method so that the dimension of the discretized problem was 288 with 14 couples of nodes on the contact interfaces. This problem was first considered admitting vertical rigid motion of the upper cylinder only. A second formulation admitted rigid body motion of both cylinders. The Lagrange multipliers of the solution are the contact nodal forces. To solve the problem with relative precision equal to 10^{-4} , three (SB) problems were solved with $\rho = 10^6$, $M = 10^4$ and $\Gamma = 0.1$. The total number of QUACAN iterations was 42.

PROBLEM 3. Finally, we consider a problem of equilibrium of a system of elastic blocks. This problem arises in mining engineering. An example of the solution of such problems under the assumption of plane strain may be found in [7]. The difficulties related to the analysis of equilibrium of block structures comprise identification of unknown contact interface, necessity to deal with floating blocks that do not have enough boundary conditions and often large matrices that arise from the finite element discretization of 3D problems. To test the performance of our algorithm we solved a 3D problem proposed by Hittinger in [27]. The 2D version of this problem was solved in [27] and [8]. A description of the problem and the variants solved with our algorithm are in [12]. Main characteristics of the two block variant were 4663 nodal variables and 221 dual variables (unknown contact nodal forces), while the three block variant comprised 6419 nodal variables and 382 dual variables. The bandwidth was 165 in both variants. The solution to relative precision 10^{-4} was obtained with three outer iterations, that is, just three (SB) problems were necessary. The number of inner QUACAN conjugate gradient iterations for two and three block problems was 105 and 276, respectively.

7. Comments and conclusions

We have described a new algorithm for the solution of coercive and semicoercive contact problems of elasticity without friction based on variational formulation reduced to the boundary. The method directly obtains the tractions on the contact interface. The stress and strain distribution may then be obtained by the solution of standard linear problems for each body separately.

The algorithm combines a variant of the domain decomposition method of the Neumann-Neumann type based on the duality theory of quadratic programming with the new algorithms for the solution of the quadratic programming problems with simple bounds. For the solution of semicoercive problems, these methods are exploited in the augmented Lagrangian algorithm. A new feature of these algorithms is the adaptive control of precision of the solution of auxiliary problems with effective usage of the projections and penalty technique.

The implementation of the algorithm deals separately with each body, so that the algorithm is suitable for parallel implementation. First numerical experiments indicate that the algorithms presented are efficient. We believe that the performance of the algorithms may be considerably improved by the ‘coarse grid’ preconditioner in combination with the standard regular preconditioners as presented at this conference by F.-X. Roux et al. [29].

References

1. O. Axelsson, *Iterative solution methods*, Cambridge University Press, Cambridge, 1994.
2. O. Axelsson and G. Lindskog, *On the rate of convergence of the preconditioned conjugate gradient method*, Num. Math. **48** (1986), 499–523.
3. R. H. Bielschowsky, A. Friedlander, F. A. M. Gomes, J. M. Martínez, and M. Raydan, *An adaptive algorithm for bound constrained quadratic minimization*, To appear in *Investigación Oper.*, 1995.
4. I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek, *Solution of variational inequalities in mechanics*, Springer Verlag, Berlin, 1988.
5. A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Num. Anal. **28** (1991), 545–572.
6. ———, *Lancelot: a fortran package for large scale nonlinear optimization*, Springer Verlag, Berlin, 1992.
7. Z. Dostál, *Numerical modelling of jointed rocks with large contact zone*, Mech. of Jointed Rocks (M. P. Rosmanith, ed.), A. A. Balkema, Rotterdam, 1990, pp. 595–597.
8. ———, *Conjugate projector preconditioning for the solution of contact problems*, Internat. J. Numer. Methods Engrg. **34** (1992), 271–277.
9. ———, *Duality based domain decomposition with inexact subproblem solver for contact problems*, Contact Mechanics II (M. H. Alibadi and C. Alessandri, eds.), Wessex Inst. of Technology, Southampton, 1995, pp. 461–468.
10. ———, *Duality based domain decomposition with proportioning for the solution of free-boundary problems*, J. Comput. Appl. Math. **63** (1995), 203–208.
11. ———, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Opt. **7** (1997), 871–887.
12. Z. Dostál, A. Friedlander, and S. A. Santos, *Analysis of block structures by augmented Lagrangians with adaptive precision control*, To appear in *Proceedings of GEOMECHANICS'96*, A. A. Balkema, Rotterdam, 1996.
13. ———, *Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints*, Technical Report RP 74/96, Institute of Mathematics, Statistics and Scientific Computation, University of Campinas,, October 1996.
14. ———, *Analysis of semicoercive contact problems using symmetric BEM and augmented Lagrangians*, To appear in *Eng. Anal. Bound. El.*, 1997.

15. ———, *Solution of contact problems using subroutine box-quacan*, To appear in *Investigación Oper.*, 1997.
16. Z. Dostál and V. Vondrák, *Duality based solution of contact problems with Coulomb friction*, Arch. Mech. **49** (1997), 453–460.
17. F.-X. Roux et. al., *Spectral analysis of interface operator*, Proceedings of the 5th Int. Symp. on Domain Decomposition Methods for Partial Differential Equations (D. E. Keyes et al., ed.), SIAM, Philadelphia, 1992, pp. 73–90.
18. C. Farhat, P. Chen, and F.-X. Roux, *The dual Schur complement method with well posed local Neumann problems*, SIAM J. Sci. Stat. Comput. **14** (1993), 752–759.
19. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Eng. **115** (1994), 365–385.
20. C. Farhat and F.-X. Roux, *An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems*, SIAM J. Sci. Stat. Comput. **13** (1992), 379–396.
21. A. Friedlander and J. M. Martínez, *On the maximization of concave quadratic functions with box constraints*, SIAM J. Opt. **4** (1994), 177–192.
22. A. Friedlander, J. M. Martínez, and M. Raydan, *A new method for large scale box constrained quadratic minimization problems*, Optim. Methods Softw. **5** (1995), 57–74.
23. A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim. **30** (1994), 235–266.
24. ———, *On the resolution of large-scale linearly constrained convex minimization problems*, SIAM J. Opt. **4** (1994), 331–339.
25. ———, *Solution of linear complementarity problems using minimization with simple bounds*, J. Global Optim. **6** (1995), 253–267.
26. ———, *A strategy for solving variational inequalities using minimization with simple bounds*, Numer. Funct. Anal. Optim. **16** (1995), 653–668.
27. M. Hittinger, *Numerical analysis of toppling failures in jointed rocks*, Ph.D. thesis, University of California, Department of Civil Engineering, 1978.
28. N. Kikuchi and J. T. Oden, *Contact problems in elasticity*, SIAM, Philadelphia, 1988.
29. F.-X. Roux, *Efficient parallel ‘coarse grid’ preconditioner to the FETI method*, This conference, 1997.

DEPARTMENT OF APPLIED MATHEMATICS, VŠB-TECHNICAL UNIVERSITY OSTRAVA, TR 17.
 LISTOPADU, CZ-70833 OSTRAVA, CZECH REPUBLIC
E-mail address: zdenek.dostal@vsb.cz

DEPARTMENT OF APPLIED MATHEMATICS, IMECC – UNICAMP, UNIVERSITY OF CAMPINAS,
 CP 6065, 13081-970 CAMPINAS SP, BRAZIL
E-mail address: friedlan@ime.unicamp.br

DEPARTMENT OF MATHEMATICS, IMECC – UNICAMP, UNIVERSITY OF CAMPINAS, CP 6065,
 13081-970 CAMPINAS SP, BRAZIL
E-mail address: sandra@ime.unicamp.br

An Iterative Substructuring Method for Elliptic Mortar Finite Element Problems with Discontinuous Coefficients

Maksymilian Dryja

1. Introduction

In this paper, we discuss a domain decomposition method for solving linear systems of algebraic equations arising from the discretization of elliptic problem in the 3-D by the mortar element method, see [4, 5] and the literature given therein. The elliptic problem is second-order with piecewise constant coefficients and the Dirichlet boundary condition. Using the framework of the mortar method, the problem is approximated by a finite element method with piecewise linear functions on nonmatching meshes.

Our domain decomposition method is an iterative substructuring one with a new coarse space. It is described as an additive Schwarz method (ASM) using the general framework of ASMs; see [11, 10]. The method is applied to the Schur complement of our discrete problem, i.e. we assume that interior variables of all subregions are first eliminated using a direct method.

In this paper, the method is considered for the mortar elements in the geometrically conforming case, i.e. the original region Ω , which for simplicity of presentation is a polygonal region, is partitioned into polygonal subregions (substructures) Ω_i that form a coarse finite element triangulation.

The described ASM uses a coarse space spanned by special functions associated with the substructures Ω_i . The remaining spaces are local and are associated with the mortar faces of the substructures and the nodal points of the wire basket of the substructures. The problems in these subspaces are independent so the method is well suited for parallel computations. The described method is almost optimal and its rate of convergence is independent of the jumps of coefficients.

The described method is a generalization of the method presented in [8] to second order elliptic problems with discontinuous coefficients. Other iterative substructuring methods for the mortar finite elements have been described and analyzed in several papers, see [1, 2, 6, 12, 13] and the literature given therein. Most of them are devoted to elliptic problems with regular coefficients and the 2-D case.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N30, 65N22, 65N10.

The author was supported in part by the National Science Foundation under Grant NSF-CCR-9503408 and Polish Science Foundation under Grant 102/P03/95/09.

The outline of the paper is as follows: In Section 2, the discrete problem obtained from the mortar element technique in the geometrically conforming case is described. In Section 3, the method is described in terms of an ASM and Theorem 1 is formulated as the main result of the paper. A proof of this theorem is given in Section 5 after that certain auxiliary results, which are needed for that proof, are given in Section 4.

2. Mortar discrete problem

We solve the following differential problem: Find $u^* \in H_0^1(\Omega)$ such that

$$(1) \quad a(u^*, v) = f(v), \quad v \in H_0^1(\Omega),$$

where

$$a(u, v) = \sum_{i=1}^N \rho_i (\nabla u, \nabla v)_{L^2(\Omega_i)}, \quad f(v) = (f, v)_{L^2(\Omega)},$$

$\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$ and ρ_i is a positive constant.

Here Ω is a polygonal region in the 3-D and the Ω_i are polygonal subregions of diameter H_i . They form a coarse triangulation with a mesh parameter $H = \max_i H_i$. In each Ω_i triangulation is introduced with triangular elements $e_j^{(i)}$ and a parameter $h_i = \max_j h_i^{(j)}$ where $h_i^{(j)}$ is a diameter of $e_i^{(j)}$. The resulting triangulation of Ω can be nonmatching. We assume that the coarse triangulation and the h_i -triangulation in each Ω_i are shape-regular in the sense of [7]. 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

$$X^h(\Omega) = X_1(\Omega_1) \times \cdots \times X_N(\Omega_N).$$

To define the mortar finite element method, we introduce some notation and spaces. Let

$$\Gamma = (\cup_i \partial\Omega_i) \setminus \partial\Omega$$

and let F_{ij} and E_{ij} denote the faces and edges of Ω_i . The union of \bar{E}_{ij} forms the wire basket W_i of Ω_i . We now select open faces γ_m of Γ , called mortars (masters), such that

$$\bar{\Gamma} = \cup_m \bar{\gamma}_m \text{ and } \gamma_m \cap \gamma_n = \emptyset \text{ if } m \neq n.$$

We denote the face of Ω_i by $\gamma_{m(i)}$. Let $\gamma_{m(i)} = F_{ij}$ be a face common to Ω_i and Ω_j . F_{ij} as a face of Ω_j is denoted by $\delta_{m(j)}$ and it is called nonmortar (slave). The rule for selecting $\gamma_{m(i)} = F_{ij}$ as mortar is that $\rho_i \geq \rho_j$. Let $W^{h_i}(F_{ij})$ be the restriction of $X_i(\Omega_i)$ to F_{ij} . Note that on $F_{ij} = \gamma_{m(i)} = \delta_{m(j)}$ we have two triangulation and two different face spaces $W^{h_i}(\gamma_{m(i)})$ and $W^{h_j}(\delta_{m(j)})$.

Let $M^{h_j}(\delta_{m(j)})$ denote a subspace of $W^{h_j}(\delta_{m(j)})$ defined as follows: The values at interior nodes of $\delta_{m(j)}$ are arbitrary, while those at nodes on $\partial\delta_{m(j)}$ are a convex combination values at interior neighboring nodes:

$$v(x_k) = \sum_{i=1}^{n_k} \alpha_i v(x_{i(k)}) \varphi_{i(k)}, \quad \sum_{i=1}^{n_k} \alpha_i = 1.$$

Here $\alpha_i \geq 0$, $x_k \in \partial\delta_{m(j)}$ and the sum is taken over interior nodal points $x_{i(k)}$ of $\delta_{m(j)}$ such that an interval $(x_k, x_{i(k)})$ is an edge of the triangulation and their number is equal to n_k ; $\varphi_{i(k)}$ is a nodal basis function associated with $x_{i(k)}$, for details see [4].

We say that $u_{i(m)}$ and $u_{j(m)}$, the restrictions of $u_i \in X_i(\Omega_i)$ and $u_j \in X_j(\Omega_j)$ to δ_m , a face common to Ω_i and Ω_j , satisfy the mortar condition if

$$(2) \quad \int_{\delta_m} (u_{i(m)} - u_{j(m)}) w ds = 0, \quad w \in M^{h_j}(\delta_m).$$

This condition can be rewritten as follows: Let $\Pi_m(u_{i(m)}, v_{j(m)})$ denote a projection from $L^2(\delta_m)$ on $W^{h_j}(\delta_m)$ defined by

$$(3) \quad \int_{\delta_m} \Pi_m(u_{i(m)}, v_{j(m)}) w ds = \int_{\delta_m} u_{i(m)} w ds, \quad w \in M^{h_j}(\delta_m)$$

and

$$(4) \quad \Pi_m(u_{i(m)}, v_{j(m)})|_{\partial\delta_m} = v_{j(m)}.$$

Thus $u_{j(m)} = \Pi_m(u_{i(m)}, v_{j(m)})$ if $v_{j(m)} = u_{j(m)}$ on $\partial\delta_m$.

By V^h we denote a space of $v \in X^h$ which satisfy the mortar condition for each $\delta_m \subset \Gamma$. The discrete problem for (1) in V^h is defined as follows: Find $u_h^* \in V^h$ such that

$$(5) \quad a(u_h^*, v_h) = f(v_h), \quad v_h \in V^h,$$

where

$$a(u_h, v_h) = \sum_{i=1}^N a_i(u_{ih}, v_{ih}) = \sum_{i=1}^N \rho_i(\nabla u_{ih}, \nabla v_{ih})_{L^2(\Omega_i)}$$

and $v_h = \{v_{ih}\}_{i=1}^N \in V^h$. V^h is a Hilbert space with an inner product defined by $a(u, v)$. This problem has an unique solution and an estimate of the error is known, see [4].

We now give a matrix form of (5). Let

$$V^h = \text{span}\{\Phi_k\}$$

where $\{\Phi_k\}$ are mortar basis functions associated with interior nodal points of the substructures Ω_i and the mortars $\gamma_{m(i)}$, and with nodal points of $\partial\gamma_{m(i)}$ and $\partial\delta_{m(i)}$, except those on $\partial\Omega$. These sets of nodal points are denoted by adding the index h . The functions Φ_k are defined as follows. For $x_k \in \Omega_{ih}$, $\Phi_k(x) = \varphi_k(x)$, the standard nodal basis function associated with x_k . For $x_k \in \gamma_{m(i)h}$, $\Phi_k = \varphi_k$ on $\gamma_{m(i)} \subset \partial\Omega_i$ and $\Pi_m(\varphi_k, 0)$ on $\delta_{m(j)} = \gamma_{m(i)} \subset \partial\Omega_j$, see (3) and (4), and $\Phi_k = 0$ at the remaining nodal points. If x_k is a nodal point common to two or more boundaries of mortars $\gamma_{m(i)}$, then $\Phi_k(x) = \varphi_k$ on these mortars and extended on the nonmortars $\delta_{m(j)}$ by $\Pi_m(\varphi_k, 0)$, and set to zero at the remaining nodal points. Let x_k be a common nodal point to two or more boundaries of nonmortars $\delta_{m(j)}$, then $\Phi_k = \Pi_m(0, \varphi_k)$ on these nonmortars and zero at the remaining nodal points. In the case when x_k is a common nodal point to boundaries of mortars and nonmortars faces, Φ_k is defined on these faces as above. Note that there are no basis functions associated with interior nodal points of the nonmortar faces.

Using these basis functions, the problem (5) can be rewritten as

$$(6) \quad A\mathbf{u}_h^* = \mathbf{f}$$

where \mathbf{u}_h^* is a vector of nodal values of u_h^* . The matrix is symmetric and positive definite, and its condition number is similar to that of a conforming finite element method provided that the h_i are all of the same order.

3. The additive Schwarz method

In this section, we describe an iterative substructuring method in terms of an additive Schwarz method for solving (5). It will be done for the Schur complement system. For that we first eliminate all interior unknowns of Ω_i using for $u_i \in X_i(\Omega_i)$ the decomposition $u_i = P u_i + H u_i$. Here and below, we drop the index h for functions. $H u_i$ is discrete harmonic in Ω_i in the sense of $(\nabla u_i, \nabla v_i)_{L^2(\Omega_i)}$ with $H u_i = u_i$ on $\partial\Omega_i$. We obtain

$$(7) \quad s(u^*, v) = f(v), \quad v \in V^h$$

where from now on V^h denote the space of piecewise discrete harmonic functions and

$$s(u, v) = a(u, v), \quad u, v \in V^h.$$

An additive Schwarz method for (7) is designed and analyzed using the general ASM framework, see [11], [10]. Thus, the method is designed in terms of a decomposition of V^h , certain bilinear forms given on these subspaces, and the projections onto these subspaces in the sense of these bilinear forms.

The decomposition of V^h is taken as

$$(8) \quad V^h(\Omega) = V_0(\Omega) + \sum_{\gamma_m \subset \Gamma} V_m^{(F)}(\Omega) + \sum_{i=1}^N \sum_{x_k \in W_{ih}} V_k^{(W_i)}(\Omega).$$

The space $V_m^{(F)}(\Omega)$ is a subspace of V^h associated with the master face γ_m . Any function of $V_m^{(F)}$ differs from zero only on γ_m and δ_m . W_{ih} is the set of nodal points of W_i and $V_k^{(W_i)}$ is an one-dimensional space associated with $x_k \in W_{ih}$ and spanned by Φ_k .

The coarse space V_0 is spanned by discrete harmonic functions Ψ_i defined as follows. Let the set of substructures Ω_i be partitioned into two sets N_I and N_B . The boundary of a substructure in N_B intersects $\partial\Omega$ in at least one point, while those of the interior set N_I , do not. For simplicity of presentation, we assume that $\partial\Omega_i \cap \partial\Omega$ for $i \in N_B$ are faces. The general case when $\partial\Omega_i \cap \partial\Omega$ for $i \in N_B$ are also edges and vertices, can be analyzed as in [10]. The function Ψ_i is associated with Ω_i for $i \in N_I$ and it is defined by its values on boundaries of substructures as follows: $\Psi_i = 1$ on $\bar{\gamma}_{m(i)} \subset \partial\Omega_i$, the mortar faces of Ω_i , and $\Psi_i = \Pi_m(1, 0)$ on $\delta_{m(j)} = \gamma_{m(i)}$, the face common to Ω_i and Ω_j ; see (3) and (4). On the nonmortar faces $\bar{\delta}_{m(i)} \subset \partial\Omega_i$, $\Psi_i = \Pi_m(0, 1)$. It is zero on the remaining mortar and nonmortar faces. We set

$$(9) \quad V_0 = \text{span}\{\Psi_i\}_{i \in N_I}.$$

Let us now introduce bilinear forms defined on the introduced spaces. $b_m^{(F)}$ associated with $V_m^{(F)} \times V_m^{(F)} \rightarrow R$, is of the form

$$(10) \quad b_m^{(F)}(u_{m(i)}, v_{m(i)}) = \rho_i(\nabla u_{m(i)}, \nabla v_{m(i)})_{L^2(\Omega_i)},$$

where $u_{m(i)}$ is the discrete harmonic function in Ω_i with data $u_{m(i)}$ on the mortar face $\gamma_{m(i)}$ of Ω_i , which is common to Ω_j , and zero on the remaining faces of Ω_i .

We set $b_k^{(W_i)} : V_k^{(W_i)} \times V_k^{(W_i)} \rightarrow R$, equals to $a(u, v)$.

A bilinear form $b_0(u, v) : V_0 \times V_0 \rightarrow R$, is of the form

$$(11) \quad \begin{aligned} b_0(u, v) = & \sum_{i \in N_I} (1 + \log \frac{H_i}{h_i}) H_i \rho_i \sum_{\delta_{m(i)} \subset \partial \Omega_i} (\alpha_j \bar{u}_j - \bar{u}_i)(\alpha_j \bar{v}_j - \bar{v}_i) + \\ & + \sum_{i \in N_B} (1 + \log \frac{H_i}{h_i}) H_i \rho_i \sum_{\delta_{m(i)} \subset \partial \Omega_i} \bar{u}_j \bar{v}_j. \end{aligned}$$

Here $\delta_{m(i)} = \gamma_{m(j)}$ is the face common to Ω_i and Ω_j , $\alpha_j = 0$ if $\delta_{m(i)} = \gamma_{m(j)} \subset \partial \Omega_j$ and $j \in N_B$, otherwise $\alpha_j = 1$,

$$(12) \quad u = \sum_{i \in N_I} \bar{u}_i \Psi_i, \quad v = \sum_{i \in N_I} \bar{v}_i \Psi_i$$

and \bar{u}_i is the discrete average value of u_i over $\partial \Omega_{ih}$, i.e.

$$(13) \quad \bar{u}_i = \left(\sum_{x \in \partial \Omega_{ih}} u_i(x) \right) / m_i,$$

and m_i is the number of nodal points of $\partial \Omega_{ih}$.

Let us now introduce operators $T_m^{(F)}$, $T_k^{(W_i)}$ and T_0 by the bilinear forms $b_m^{(F)}$, $b_k^{(W_i)}$ and b_0 , respectively, in the standard way. For example, $T_m^{(F)} : V^h \rightarrow V_m^{(F)}$, is the solution of

$$(14) \quad b_m^{(F)}(T_m^{(F)} u, v) = a(u, v), \quad v \in V_m^{(F)}.$$

Let

$$T = T_0 + \sum_{\gamma_m \subset \Gamma} T_m^{(F)} + \sum_{i=1}^N \sum_{x_k \in W_{ih}} T_k^{(W_i)}.$$

The problem (5) is replaced by

$$(15) \quad Tu^* = g$$

with the appropriate right-hand side.

THEOREM 1. *For all $u \in V^h$*

$$(16) \quad C_0 (1 + \log \frac{H}{h})^{-2} a(u, u) \leq a(Tu, u) \leq C_1 a(u, u)$$

where C_i are positive constants independent of $H = \max_i H_i$, $h = \min_i h_i$ and the jumps of ρ_i .

4. Auxiliary results

In this section, we formulate some auxiliary results which we need to prove Theorem 1.

Let for $u \in V^h$, $u_0 \in V_0$ be defined as

$$(17) \quad u_0 = \sum_{i \in N_I} \bar{u}_i \Psi_i$$

where the \bar{u}_i are defined in (13).

LEMMA 2. *For $u_0 \in V_0$ defined in (17)*

$$(18) \quad a(u_0, u_0) \leq C b_0(u_0, u_0)$$

where $b_0(., .)$ is given in (11) and C is a positive constant independent of the H_i , h_i and the jumps of ρ_i .

PROOF. Note that u_0 on $\partial\Omega_i$, $i \in N_I$, is of the form

$$(19) \quad u_0 = \bar{u}_i \Psi_i + \sum_j \bar{u}_j \Psi_j$$

where the sum is taken over the nonmortars $\delta_{m(i)} = \gamma_{m(j)}$ of Ω_i and $\gamma_{m(j)}$ is the face common to Ω_i and Ω_j . In this formula $\Psi_j = 0$ if $j \in N_B$. Let us first discuss the case when all $j \in N_I$ in (19). Note that $\Psi_i + \sum_j \Psi_j = 1$ on $\partial\Omega_i$. Using this, we have

$$\rho_i |u_0|_{H^1(\Omega_i)}^2 = \rho_i |u_0 - \bar{u}_i|_{H^1(\Omega_i)}^2 \leq C \sum_{\delta_{m(i)} \subset \partial\Omega_i} \rho_i (\bar{u}_j - \bar{u}_i)^2 \|\Psi_j\|_{H_{00}^{\frac{1}{2}}(\delta_{m(i)})}^2.$$

It can be shown that

$$(20) \quad \|\Psi_j\|_{H_{00}^{\frac{1}{2}}(\delta_{m(i)})}^2 \leq CH_i (1 + \log \frac{H_i}{h_i}).$$

For that note that $\Psi_j = \Pi_m(1, 0)$ on $\delta_{m(i)}$ and use the properties of Π_m ; for details see the proof of Lemma 4.5 in [8]. Thus

$$(21) \quad \rho_i |u_0|_{H^1(\Omega_i)}^2 \leq CH_i \sum_{\delta_{m(i)} \subset \partial\Omega_i} \rho_i (1 + \log \frac{H_i}{h_i}) (\bar{u}_j - \bar{u}_i)^2.$$

For $i \in N_I$ with $j \in N_B$, we have

$$\rho_i |u_0|_{H^1(\Omega_i)}^2 \leq CH_i \sum_{\delta_{m(i)} \subset \partial\Omega_i} \rho_i (1 + \log \frac{H_i}{h_i}) (\alpha_j \bar{u}_j - \bar{u}_i)^2$$

where $\alpha_j = 0$ if $\delta_{m(i)} = \gamma_{m(j)} \subset \partial\Omega_j$ and $j \in N_B$, otherwise $\alpha_j = 1$. For $i \in N_B$

$$\rho_i |u_0|_{H^1(\Omega_i)}^2 \leq CH_i \sum_{\delta_{m(i)} \subset \partial\Omega_i} \rho_i (1 + \log \frac{H_i}{h_i}) \bar{u}_j^2$$

Summing these inequalities with respect to i , we get

$$\begin{aligned} a(u_0, u_0) &\leq C \left\{ \sum_{i \in N_I} (1 + \log \frac{H_i}{h_i}) H_i \rho_i \sum_{\delta_{m(i)} \subset \partial\Omega_i} (\alpha_j \bar{u}_j - \bar{u}_i)^2 + \right. \\ &\quad \left. + \sum_{i \in N_B} (1 + \log \frac{H_i}{h_i}) H_i \rho_i \sum_{\delta_{m(i)} \subset \partial\Omega_i} \bar{u}_j^2 \right\}, \end{aligned}$$

which proves (18). \square

LEMMA 3. Let $\gamma_{m(i)} = \delta_{m(j)}$ be the face common to Ω_i and Ω_j , and let $u_{i(m)}$ and $u_{j(m)}$ be the restrictions of $u_i \in X_i(\Omega_i)$ and $u_j \in X_j(\Omega_j)$ to $\gamma_{m(i)}$ and $\delta_{m(j)}$, respectively. Let $u_{i(m)}$ and $u_{j(m)}$ satisfy the mortar condition (2) on $\delta_{m(j)}$. If $u_{i(m)}$ and $u_{j(m)}$ vanish on $\partial\gamma_{m(i)}$ and $\partial\delta_{m(j)}$, respectively, then

$$\|u_{j(m)}\|_{H_{00}^{\frac{1}{2}}(\delta_{m(j)})}^2 \leq C \|u_{i(m)}\|_{H_{00}^{\frac{1}{2}}(\gamma_{m(i)})}^2$$

where C is independent of h_i and h_j .

This lemma follows from Lemma 1 in [3]. A short proof for our case is given in Lemma 4.2 of [8].

LEMMA 4. Let Φ_k be a function defined in Section 2 and associated with a nodal point $x_k \in W_i \subset \partial\Omega_i$. Then

$$a(\Phi_k, \Phi_k) \leq Ch_i \rho_i \sum_{\gamma_{m(i)} \subset \partial\Omega_i} (1 + \log \frac{h_i}{h_j})$$

where C is independent of h_i and ρ_i , and $\gamma_{m(i)} = \delta_{m(j)}$.

The proof of this lemma differs slightly from that of Lemma 4.3 in [8], therefore it is omitted here.

5. Proof of Theorem 1

Using the general theorem of ASMs, we need to check three key assumptions; see [11] and [10].

Assumption (iii) For each $x \in \Omega$ the number of substructures with common x is fixed, therefore $\rho(\varepsilon) \leq C$.

Assumption (ii) Of course $\omega = 1$ for $b_k^{(W_i)}(u, u)$, $u \in V_k^{(W_i)}$. The estimate

$$a(u, u) \leq \omega b_0(u, u), \quad u \in V_0$$

follows from Lemma 2 with $\omega = C$.

We now show that for $u \in V_m^{(F)}$, see (10),

$$(22) \quad a(u, u) \leq C b_m^{(F)}(u, u).$$

Let $\gamma_{i(m)} = \delta_{j(m)}$ be the mortar and nonmortar sides of Ω_i and Ω_j , respectively. For $u \in V_m^{(F)}$, we have

$$a(u, u) = a_i(u_i, u_i) + a_j(u_j, u_j) \leq C(\rho_i \|u_i\|_{H_{00}^{\frac{1}{2}}(\gamma_{i(m)})}^2 + \rho_j \|u_j\|_{H_{00}^{\frac{1}{2}}(\delta_{j(m)})}^2).$$

Using now Lemma 3 and the fact that $\rho_i \geq \rho_j$ since $\gamma_{m(i)}$ is the mortar, we get (22), i.e. $\omega = C$.

Assumption (i) We show that for $u \in V^h$, there exists a decomposition

$$(23) \quad u = u_0 + \sum_{\gamma_m \subset \Gamma} u_m^{(F)} + \sum_{i=1}^N \sum_{x_k \in W_{ih}} u_k^{(W_i)},$$

where $u_0 \in V_0$, $u_m^{(F)} \in V_m^{(F)}$ and $u_k^{(W_i)} \in V_k^{(W_i)}$, such that

$$(24) \quad \begin{aligned} b_0(u_0, u_0) + \sum_{\gamma_m \subset \Gamma} b_m^{(F)}(u_m^{(F)}, u_m^{(F)}) + \sum_{i=1}^N \sum_{x_k \in W_{ih}} b_k^{(W_i)}(u_k^{(W_i)}, u_k^{(W_i)}) \\ \leq C(1 + \log \frac{H}{h})^2 a(u, u). \end{aligned}$$

Let u_0 be defined by (17), and let w_i be the restriction of $w = u - u_0$ to $\bar{\Omega}_i$. It is decomposed on $\partial\Omega_i$ as

$$(25) \quad w_i = \sum_{F_{ij} \subset \partial\Omega_i} w_i^{(F_{ij})} + w_i^{(W_i)}, \quad w_i^{(W_i)} = \sum_{x_k \in W_{ih}} w_i(x_k) \Phi_k$$

where $w_i^{(F_{ij})}$ is the restriction of $w_i - w_i^{(W_i)}$ to F_{ij} , the face of Ω_i , and zero on $\partial\Omega_i \setminus F_{ij}$.

To define $u_m^{(F)}$, let $F_{ij} = \gamma_{m(i)} = \delta_{m(j)}$ be a face common to Ω_i and Ω_j . We set

$$u_m^{(F)} = \{w_i^{(F_{ij})} \text{ on } \partial\Omega_i \text{ and } w_j^{(F_{ij})} \text{ on } \partial\Omega_j\}$$

and set it to zero at the remaining nodal points of Γ . The function $u_k^{(W_i)}$ is defined as

$$(26) \quad u_k^{(W_i)}(x) = w_i(x_k) \Phi_k(x).$$

It is easy to see that these functions satisfy (23).

To prove (24), we first show that

$$(27) \quad b_0(u_0, u_0) \leq C(1 + \log \frac{H}{h})a(u, u).$$

Note that, see (11), for $\delta_{m(i)} \subset \delta\Omega_i$, $i \in N_I$ with $j \in N_I$ when $\delta_{m(i)} = F_{ij}$ is a face common to Ω_i and Ω_j ,

$$H_i \rho_i (\bar{u}_j - \bar{u}_i)^2 \leq CH_i^{-1} \{ \rho_i \|u_i\|_{L^2(\partial\Omega_i)}^2 + \rho_j \|u_j\|_{L^2(\partial\Omega_j)}^2 \}.$$

Using the fact that the average values of u_j and u_i over $\delta_{m(i)} = \gamma_{m(j)} = F_{ij}$ are equal to each other, and using the Poincare inequality, we get

$$H_i \rho_i (\bar{u}_j - \bar{u}_i)^2 \leq C \{ \rho_i |u_i|_{H^1(\Omega_i)}^2 + \rho_j |u_j|_{H^1(\Omega_j)}^2 \}.$$

For $i \in N_I$ with $j \in N_B$ we have similar estimates:

$$H_i \rho_i (\alpha_j \bar{u}_j - \bar{u}_i)^2 \leq C \{ \rho_i |u_i|_{H^1(\Omega_i)}^2 + \rho_j |u_j|_{H^1(\Omega_j)}^2 \}.$$

Here we have used the Friedrichs inequality in Ω_j . Thus

$$(28) \quad \sum_{i \in N_I} \sum_{\delta_{m(i)} \subset \partial\Omega_i} H_i \rho_i (\alpha_j \bar{u}_j - \bar{u}_i)^2 \leq Ca(u, u).$$

In the similar way it is shown that for $i \in N_B$

$$H_i \rho_i \bar{u}_j^2 \leq C \{ \rho_i |u_i|_{H^1(\Omega_i)}^2 + \rho_j |u_j|_{H^1(\Omega_j)}^2 \}.$$

Summing this with respect to $i \in N_B$ and adding the resulting inequality to (28), we get (27).

Let us now consider the estimate for $u_m^{(F)} \in V_M^{(F)}$ when $\gamma_{m(i)} = \delta_{m(j)} = F_{ij}$, the face common to Ω_i and Ω_j . We have, see (10),

$$b_m^{(F)}(u_m^{(F)}, u_m^{(F)}) \leq C \rho_i \|w_i^{(F_{ij})}\|_{H_{00}^{\frac{1}{2}}(\gamma_{m(i)})}^2.$$

Note that on $F_{ij} = \gamma_{m(i)}$

$$w_i^{(F_{ij})} = I_{h_i}(\theta_{F_{ij}} u_i) - I_{h_i}(\theta_{F_{ij}} u_0)$$

where $\theta_{F_{ij}} = 1$ at interior nodal points of the h_i -triangulation of F_{ij} and zero on ∂F_{ij} , and I_{h_i} is the interpolant. Using Lemma 4.5 from [9], we have

$$\|I_{h_i}(\theta_{F_{ij}} u_i)\|_{H_{00}^{\frac{1}{2}}(F_{ij})}^2 \leq C(1 + \log \frac{H_i}{h_i})^2 \|u_i\|_{H^1(\Omega_i)}^2.$$

To estimate the second term, note that $u_0 = \bar{u}_i \Psi_i = \bar{u}_i$ on \bar{F}_{ij} since it is the mortar. Using Lemma 4.4 from [9], we get

$$\begin{aligned} \|I_h(\theta_{F_{ij}} u_0)\|_{H_{00}^{\frac{1}{2}}(F_{ij})}^2 &= (\bar{u}_i)^2 \|I_{h_i} \theta_{F_{ij}}\|_{H_{00}^{\frac{1}{2}}(F_{ij})}^2 \leq \\ &\leq CH_i^{-1} (1 + \log \frac{H_i}{h_i}) \|u_i\|_{L^2(\partial\Omega_i)}^2. \end{aligned}$$

Thus

$$\|w_i^{(F_{ij})}\|_{H_{00}^{\frac{1}{2}}(\gamma_{m(i)})}^2 \leq C \{ (1 + \log \frac{H_i}{h_i})^2 \|u_i\|_{H^1(\Omega_i)}^2 + H_i^{-1} (1 + \log \frac{H_i}{h_i}) \|u_i\|_{L^2(\partial\Omega_i)}^2 \}.$$

Using now a simple trace theorem and the Poincare inequality, we have

$$\|w_i^{(F_{ij})}\|_{H_{00}^{\frac{1}{2}}(\gamma_{m(i)})}^2 \leq C(1 + \log \frac{H_i}{h_i})^2 |u_i|_{H^1(\Omega_i)}^2.$$

Multiplying this by ρ_i and summing with respect to γ_m , we get

$$(29) \quad \sum_{\gamma_m \subset \Gamma} b_m^{(F)}(u_m^{(F)}, u_m^{(F)}) \leq C(1 + \log \frac{H}{h})^2 a(u, u).$$

We now prove that

$$(30) \quad \sum_{i=1}^N \sum_{x_k \in W_{ih}} b_k^{(W_i)}(u_k^{(W_i)}, u_k^{(W_i)}) \leq C(1 + \log \frac{H}{h})^2 a(u, u).$$

We first note that by (26) and Lemma 4

$$b_k^{(W_i)}(u_k^{(W_i)}, u_k^{(W_i)}) \leq C w_i^2(x_k) a(\Phi_k, \Phi_k) \leq C \rho_i h_i (1 + \log \frac{H_i}{h_i}) w_i^2(x_k).$$

Summing over the $x_k \in W_{ih}$, we get

$$(31) \quad \begin{aligned} \sum_{x_k \in W_{ih}} b_k^{(W_i)}(u_k^{(W_i)}, u_k^{(W_i)}) &\leq C \rho_i (1 + \log \frac{H_i}{h_i}) \{ \|u_i\|_{L^2(W_i)}^2 \\ &+ h_i \sum_{x_k \in W_{ih}} u_0^2(x_k) \}. \end{aligned}$$

Using a well known Sobolev-type inequality, see for example Lemma 4.3 in [9], we have

$$(32) \quad \|u_i\|_{L^2(W_i)}^2 \leq C(1 + \log \frac{H_i}{h_i}) \|u_i\|_{H^1(\Omega_i)}^2.$$

To estimate the second term, we note that, see (17),

$$(33) \quad h_i \sum_{x_k \in W_{ih}} u_0^2(x_k) \leq C H_i (\bar{u}_i)^2 \leq C \|u_i\|_{H^1(\Omega_i)}^2.$$

Here we have also used a simple trace theorem. Substituting (32) and (33) into (31), and using the Poincare inequality, we get

$$\sum_{x_k \in W_{ih}} b_k^{(W_i)}(u_k^{(W_i)}, u_k^{(W_i)}) \leq C(1 + \log \frac{H_i}{h_i})^2 \rho_i |u_i|_{H^1(\Omega_i)}^2.$$

Summing now with respect to i , we get (30).

To get (24), we add the inequalities (27), (29) and (30). The proof of Theorem 1 is complete.

Acknowledgments. The author is indebted to Olof Widlund for helpful comments and suggestions to improve this paper.

References

1. Y. Achdou, Yu. A. Kuznetsov, and O. Pironneau, *Substructuring preconditioner for the Q_1 mortar element method*, Numer. Math. **41** (1995), 419–449.
2. Y. Achdou, Y. Maday, and O. B. Widlund, *Iterative substructuring preconditioners for the mortar method in two dimensions*, Tech. Report 735, Courant Institute technical report, 1997.
3. F. Ben Belgacem, *The mortar finite element method with Lagrange multipliers*, Numer. Math.

4. F. Ben Belgacem and Y. Maday, *A new nonconforming approach to domain decomposition*, East-West J. Numer. Math. **4** (1994), 235–251.
5. C. Bernardi, Y. Maday, and A.T. Patera, *A new nonconforming approach to domain decomposition: The mortar element method*, College de France Seminar (H. Brezis and J.-L. Lions, eds.), Pitman, 1989.
6. M. A. Casarin and O. B. Widlund, *A hierarchical preconditioner for the mortar finite element method*, ETNA **4** (1996), 75–88.
7. P.G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam (1978).
8. M. Dryja, *An iterative substructuring method for elliptic mortar finite element problems with a new coarse space*, East-West J. Numer. Math. **5** (1997), 79–98.
9. M. Dryja, B. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1662 – 1694.
10. M. Dryja and O. B. Widlund, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math. **48** (1995), 121–155.
11. B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition. parallel multilevel methods for elliptic pdes*, Cambridge University Press, 1997.
12. P. Le Tallec, *Neumann-neumann domain decomposition algorithms for solving 2d elliptic problems with nonmatching grids*, East-West J. Numer. Math. **1** (1993), 129–146.
13. O.B. Widlund, *Preconditioners for spectral and mortar finite methods*, Eight international conference on domain decomposition methods (R. Glowinski, J. Periaux, Z. Shi, and O.B. Widlund, eds.), John Wiley and Sons, Ltd., 1996, pp. 19–32.

DEPARTMENT OF MATHEMATICS, INFORMATICS AND MECHANICS, WARSAW UNIVERSITY, BANACHA 2, 02-097 WARSAW, POLAND.

Current address: Department of Mathematics, Informatics and Mechanics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland.

E-mail address: dryja@mimuw.edu.pl.

Domain Decomposition Methods for Flow in Heterogeneous Porous Media

Magne S. Espedal, Karl J. Hersvik, and Brit G. Ersland

1. Introduction

A reservoir may consist of several different types of sediments, which in general have different porosity, ϕ , absolute permeability, \mathbf{K} , relative permeabilities, k_{ri} , and capillary pressure P_c . In this paper we will study two phase, immiscible flow, in models consisting of one or two different types of sediment. Each of the sediments may be heterogeneous. The phases are water (w) and oil (o), but the results may easily be extended to groundwater problems where the phases may be water and air or a non aqueous phase.

Such models represent a very complex and computational large problem and domain decomposition methods are a very important part of a solution procedure. It gives a good tool for local mesh refinement in regions with large gradients such as fluid fronts, interfaces between different sediments or at faults. If the models are heterogeneous, parameters in the models may be scale dependent, which means that mesh coarsening may be possible in regions with small gradients.

Let $S = S_w$ denote the water saturation. Then the incompressible displacement of oil by water in the porous media can be described by the following set of partial differential equations, given in dimensionless form:

$$(1) \quad \nabla \cdot \mathbf{u} = q_1(\mathbf{x}, t),$$

$$(2) \quad \mathbf{u} = -\mathbf{K}(\mathbf{x})M(S, \mathbf{x})\nabla p,$$

$$(3) \quad \phi(\mathbf{x})\frac{\partial S}{\partial t} + \nabla \cdot (f(S)\mathbf{u}) - \varepsilon\nabla \cdot (D(S, \mathbf{x})\nabla S) = q_2(\mathbf{x}, t).$$

Here, \mathbf{u} is the total Darcy velocity, which is the sum of the Darcy velocities of the oil and water phases:

$$\mathbf{u} = \mathbf{u}_o + \mathbf{u}_w$$

1991 *Mathematics Subject Classification*. Primary 76T05; Secondary 35M20, 65M55, 65M25.

Key words and phrases. Reservoir models, Upscaling, Hierarchical basis, Domain Decomposition, Operator splitting.

Support from The Norwegian Research Council under the program PROPETRO and from VISTA, a research cooperation between the Norwegian Academy of Science and Letters and Statoil, is gratefully acknowledged.

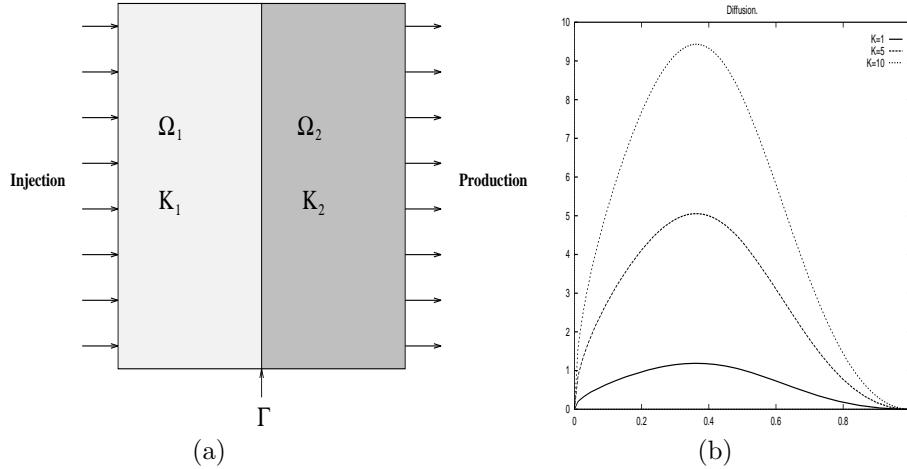


FIGURE 1. (a) Computational domain showing two different regions with different physical properties connected through an interior boundary Γ . (b) Capillary diffusion, as functions of saturation shown for permeabilities 1, 5 and 10.

Furthermore, p is the total fluid pressure given by $p = \frac{1}{2}(p_w + p_o)$, $q_i(\mathbf{x}, t)$, $i = 1, 2$ denotes contribution from the injection and production wells in addition to capillary pressure terms, which are treated explicitly in (1). $\mathbf{K}(\mathbf{x})$ is the absolute permeability, $M(S, \mathbf{x})$ denotes the nonzero total mobility of the phases and $\phi(\mathbf{x})$ is the porosity. A reasonable choice for the porosity is $\phi = 0.2$. The parameter ε is a small ($10^{-1} - 10^{-4}$) dimensionless scaling factor for the diffusion.

Equation (3) is the fractional flow formulation of the mass balance equation for water. The fractional flow function $f(S)$ is typically a S shaped function of saturation., The capillary diffusion function $D(S, \mathbf{x})$ is a bell shaped function of saturation, S , and has an almost linearly dependence of the permeability $\mathbf{K}(\mathbf{x})$. Further, $D(0, \mathbf{x}) = D(1, \mathbf{x}) = 0$. In Figure 1, a capillary diffusion function is shown for different permeabilities.

For a complete survey and justification of the model we refer to [5]. We have neglected gravity forces, but the methods described can also be applied for models where gravity effects are included [17]. The following analytical form for the capillary pressure is chosen[35]:

$$(4) \quad P_C(S, \mathbf{x}) = 0.9\phi^{-0.9}\mathbf{K}^{-0.1}\frac{1-S}{\sqrt{S}},$$

We note that the capillary pressure depend on the permeability of the rock.

The phase pressures and the capillary pressure are continuous over a boundary Γ , separating two sediments. Furthermore, the flux must be continuous [12, 14, 4, 23] which gives the following internal boundary conditions:

$$(5) \quad P_C^{\Gamma-}(S^{\Gamma-}) = P_C^{\Gamma+}(S^{\Gamma+})$$

$$(6) \quad (\mathbf{u}^{\Gamma-} - \mathbf{u}^{\Gamma+}) \cdot \mathbf{n}^\Gamma = 0$$

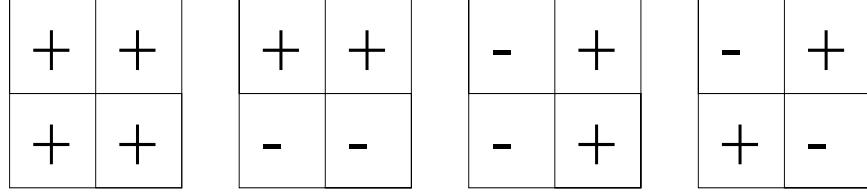


FIGURE 2. From left to right Γ_0 , Γ_α , Γ_β , Γ_γ . The + and - sign represent the functional value +1 and -1. Γ_0 is the scaling function.

(7)

$$(f(S^{\Gamma^-})\mathbf{u} - \varepsilon D^{\Gamma^-}(S^{\Gamma^-}, \mathbf{x})\nabla S^{\Gamma^-}) \cdot \mathbf{n}^\Gamma = (f(S^{\Gamma^+})\mathbf{u} - \varepsilon D^{\Gamma^+}(S^{\Gamma^+}, \mathbf{x})\nabla S^{\Gamma^+}) \cdot \mathbf{n}^\Gamma.$$

Here $(\cdot)^{\Gamma^-}$ and $(\cdot)^{\Gamma^+}$ denote the left and right hand side value at the interior boundary Γ , and \mathbf{n}^Γ is normal to the boundary.

Since the capillary pressure depends on both the saturation and the permeability, the continuity of the capillary pressure leads to discontinuous saturation over the interior boundary. Also, the flux conservation (7) creates large gradients in the saturation over an interior boundary separating two different sediments. This adds extra complexity to our problem especially when a saturation front passes an interface.

2. Representation of the Permeability

The permeability \mathbf{K} may have several orders of variation and the geometrical distribution can be very different in the sediments. Further, fractures and faults add to the complexity. Therefore, we need a good method for the representation of \mathbf{K} . Hierarchical basis may be such a tool [27, 18].

The simplicity of the Haar system makes it an attractive choice for such a representation. Further it gives a permeability representation which is consistent with a domain decomposition solution procedure. We will restrict ourselves to two space dimensions and consider the unit square as our computational domain Ω . The Haar basis of scale 0 and 1, for two space dimension, is given by the set of functions in Figure 2.

On each mesh of 2×2 cells, any cell wise constant function can be given a representation as linear combinations of Γ_0 , Γ_α , Γ_β and Γ_γ , where the coefficient of Γ_0 has a nonzero coefficient only for mesh of scale 0, which is the whole computational domain Ω . A 2-scale Haar multi resolution analysis, built on the scaling function Γ_0 [6, 9], will be used to represent the permeability.

Let the absolute permeability \mathbf{K} be a cell-wise constant function on Ω with 2^{N+1} cells in each direction.

Then \mathbf{K} can then be expanded in terms of the Haar basis:

$$(8) \quad \mathbf{K} = \mathbf{K}_0 \Gamma_0 + \sum_{i=1}^N \sum_{j=0}^{2^N} \sum_{k=0}^{2^N} \sum_{l=\alpha, \beta, \gamma} \mathbf{K}_{ijkl} \Gamma_{ijkl}$$

where the index i gives the scale and j and k give the translations of the two dimensional wavelet Γ_{ijkl} .

We should note that this representation is easily extended to three space dimensions. If the permeability is given by a tensor, the wavelet representation should be used for each of the components. Also, other types of wavelet may be used. Especially if an overlapping domain decomposition method is used, a less localised wavelet representation should be useful.

3. Solution procedure

The governing equations (1), (2) and (3) are coupled. A sequential, time marching strategy is used to decouple the equations. This strategy reflects the different natures of the elliptic pressure equation given by (1) and (2), and the convection dominated parabolic saturation equation (3). The general solution strategy is published earlier [15, 10, 25, 8, 21], so we only give the main steps in the procedure.

3.1. Sequential Solution Procedure:

- **Step 1:**

For $t \in [t_n, t_{n+1}]$, solve the pressure and velocity equations (1, 2). The velocity is a smoother function in space and time than the saturation. Therefore we linearize the pressure and velocity equations by using the saturation from the previous time step, $S = S(t_n)$. The pressure equation is solved in a weak form, using a standard Galerkin method with bilinear elements [2, 31].

Then the velocity is derived from the Darcy equation (2), using local flux conservation over the elements [31]. This gives the same accuracy for the velocity field as for the pressure. The mixed finite element method would be an alternative solution procedure. In many cases, it is not necessary to solve the pressure for each timestep.

- **Step 2:**

A good handling of the convective part of the saturation equations, is essential for a fast and accurate solution for S . We will use an operator splitting technique where an approximate saturation \tilde{S} is calculated from a hyperbolic equation:

For $t \in [t_n, t_{n+1}]$, with \mathbf{u} given from Step 1, solve:

$$(9) \quad \frac{d\tilde{S}}{d\tau} \equiv \phi \frac{\partial \tilde{S}}{\partial t} + \nabla \cdot (\bar{f}(\tilde{S})\mathbf{u}) = 0$$

where \bar{f} is a modified fractional flow function [15, 8, 21].

- **Step 3:**

The solution \tilde{S} provide a good approximation of the time derivative and the nonlinear coefficients in (3).

For $t \in [t_n, t_{n+1}]$, and $S_0 = \tilde{S}$, solve:

$$(10) \quad \frac{\partial S_i}{\partial \tau} + \nabla \cdot (b(S_{i-1})S_i\mathbf{u} - \mathbf{D}(S_{i-1}) \cdot \nabla S_i) = q_2(\mathbf{x}, t)$$

for $i = 1, 2, 3, \dots$ until convergence, where $b(S)S = f(S) - \bar{f}(S)$. Equation (10) is solved in a weak form [15, 8], using optimal test functions [3].

- **Step 4:**

Depending on the strength of the coupling between the velocity and the saturation equation, the procedure may be iterated.

3.2. Domain decomposition. The pressure (1), (2) and saturation equation (10) represent large elliptic problems. The equations are solved by using a preconditioned iterative procedure, based on domain decomposition [32]. The procedure allows for adaptive local grid refinement, which gives a better resolution at wells and in regions with large permeability variations such as interfaces between two types of sediments. The procedure also allows for adaptive local refinement at moving saturation fronts. Assuming a coarse mesh Ω_C on the computational domain Ω , we get the following algorithm [29, 8, 34, 33]:

1. Solve the splitted hyperbolic saturation equation (9) on the coarse grid Ω_C , by integrating backwards along the characteristics.
2. Identify coarse element that contain large gradients in saturation and activate refined overlapping/non overlapping sub grids Ω_k to each of these.
3. Solve equation (9) on the refined sub grids Ω_k , by integrating backwards along the characteristics.
4. Using domain decomposition methods, solve the complete saturation equation with the refined coarse elements. The characteristic solution \tilde{S} is used as the first iteration of the boundary conditions for the sub domains Ω_k .

As is well known, algorithms based on domain decompositions have good parallel properties [32].

We will present result for 2D models, but the procedure has been implemented and tested for 3D models [16].

4. Coarse Mesh Models

With the solution procedure described in Section 3, pressure, velocity and saturation will be given on a coarse mesh in some regions and on a refined mesh in the remaining part of Ω . This means that the model equations have to be given for different scales. Several upscaling techniques are given in the literature [22, 1, 11, 26, 18, 7, 30, 24]. In this paper, we will limit the analysis to the upscaling of the pressure equation. The upscaling of the saturation equation is less studied in the literature [1, 24].

The goal of all upscaling techniques is to move information from a fine grid to a coarser grid without loosing significant information along the way. A quantitative criteria is the conservation of dissipation:

$$(11) \quad e = \mathbf{K} \nabla p \cdot \nabla p$$

A related and intuitively more correct criteria, is the conservation of the mean fluid velocity $\langle \mathbf{v} \rangle$, given by:

$$(12) \quad \begin{aligned} \langle \mathbf{v} \rangle &= -\mathbf{K}_{eff} \nabla \langle p \rangle \\ \nabla \cdot \langle \mathbf{v} \rangle &= 0 \end{aligned}$$

where \mathbf{K}_{eff} is an effective permeability.

In our approach it is natural to use conservation of the mean velocity as the upscaling criteria, although the method also conserves dissipation fairly well [18].

4.1. Wavelet upscaling. A Multi Resolution Analysis (MRA), [9], defined on $L^2(R) \times L^2(R)$ consists of a nested sequence of closed subspaces of $L^2 \times L^2$ such that:

$$(13) \quad \cdots \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots$$

$$(14) \quad \bigcup_{j \in \mathbb{Z}} V_j = L^2(R) \times L^2(R)$$

$$(15) \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

$$(16) \quad V_{j+1} = V_j \cup W_j$$

$$(17) \quad f(x) \in V_j \iff f(2x) \in V_{j+1}$$

Here, W_j is the orthogonal complement of V_j in V_{j+1} . The permeability given by (8) satisfies such a hierarchical representation. Let $j = 0$ and $j = J$ be respectively the coarsest and finest scale to appear in \mathbf{K} . Upscaling in this context is to move information from scale $j = J$ to a coarser scale $j = M$, $M \leq J$.

In signal analysis, "highcut" filters are commonly used to clean a signal from its high frequency part which is considered to be negligible noise. We will apply to same reasoning on \mathbf{K} . Such an upscaling will keep all large scale information, while variation on scales below our course grid will be neglected in regions where the flow has a slow variation. Moderate fine scale changes in the permeability, give rise to heterogeneous fingers [24] and have little influence on the global transport. But this upscaling technique is only valid in the case that the coefficients on the scales $J, J-1, \dots, M$ are small compared to the coefficients on a lower scale. This would not be the case for narrow high permeability zones. However, in regions with high permeability channels, the solutions will contain large gradients, which mean that we should use a refined grid to resolve the dynamics. A coarse grid solution based on upscaling, will loose too much information in such regions.

4.2. Highcut upscaling. We will give an example of the highcut upscaling techniques on an anisotropic permeability field. Such a field is chosen because it is difficult to handle using simple mean value techniques. The coefficients which determine the permeability field at each scale are read from a file, but if we want to work with measured permeability fields, a subroutine can determine the coefficients on each scale recursively using standard two scale relations from the wavelet theory, [9].

The permeability field is generated by introducing large coefficients on a relatively low level. The goal is to model a case where we may have several regimes of permeability which on their own may be isotropic, but when put together constitute a very complex reservoir.

Figure 3 shows the permeability field for a highcut upscaling as 3-D histograms. We can clearly see the complexity and anisotropy of the permeability field. Upscaling to scale 3 gives a massive loss of information, because the scale 4 coefficient are fairly large.

Table 1 displays the coefficients used. The aspect ratio between the highest and lowest permeability is about 10 in this case [18]. The highest scale is 7 which gives $2^7 \times 2^7$ cells, and the permeability value on scale 0 is 1.0. K_α , K_β and K_γ are the coefficients of Γ_α , Γ_β and Γ_γ respectively (2).

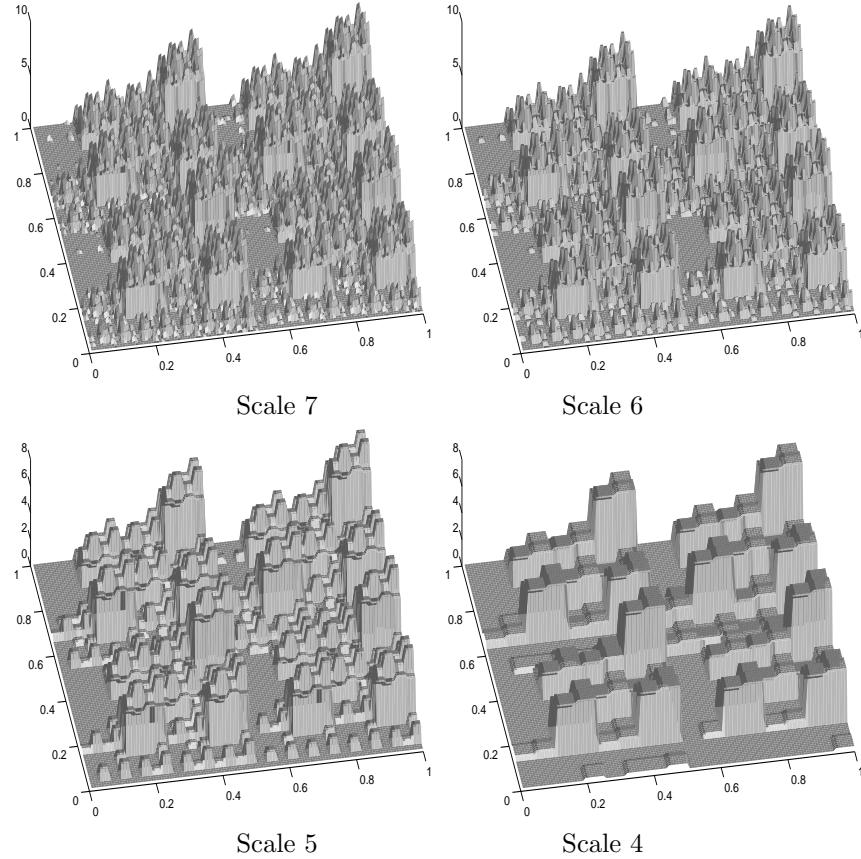


FIGURE 3. The figure shows the permeability field for a highcut upscaling in a 3-D histograms. Scale 7 → 128 × 128 cells, scale 6 → 64 × 64 cells, scale 5 → 32 × 32 cells and scale 4 → 16 × 16 cells.

TABLE 1. The table show the value of the coefficients on each level.

Level	K_α	K_β	K_γ
1	0.3	0.1	0.2
2	0.0	1.0	0.9
3	1.0	0.9	0.8
4	-0.4	0.3	0.2
5	0.5	0.2	0.3
6	0.1	-0.2	-0.3
7	0.2	0.1	0.0

5. Model error

As clearly shown in Figure 3, upscaling of permeability introduces modeling error. In order to develop a domain decomposition based simulator, the model has to be given on different scales. Therefore, we need an a posteriori error estimate of the modeling error introduced by the upscaling if we want to control the total

error in the computation. Such an error estimate can give the base for an adaptive local refinement procedure for the simulation of heterogeneous porous media.

Defining the bilinear form:

$$A : H^1(\Omega) \times H_0^1(\Omega) \mapsto R$$

$$(18) \quad A(p, \phi) = \int_{\Omega} \nabla p \cdot \mathbf{K} \nabla \phi d\mathbf{x}$$

Then, the fine scale weak form of the pressure equation (1, 2) is given by:

Find $p \in H^1(\Omega)$, such that

$$(19) \quad A(p, \phi) = (q_1, \phi), \quad \forall \phi \in H_0^1(\Omega)$$

Denoting the upscaled permeability tensor by \mathbf{K}^0 , we get the equivalent upscaled model:

Find $p^0 \in H^1(\Omega)$, such that

$$(20) \quad A^0(p^0, \phi) = (q, \phi), \quad \forall \phi \in H_0^1(\Omega)$$

The modeling error introduced by replacing the fine scale permeability tensor with an upscaled permeability tensor may be given by:

$$(21) \quad e_{model} = \|\mathbf{v} - \mathbf{v}^0\|$$

where $\|\cdot\|$ is a norm defined on $L^2(\Omega) \times L^2(\Omega)$. If we define a norm as given in [26], we can avoid the appearance of the fine scale solution on the right hand side in the error estimate :

Let $\mathbf{w} \in L^2(\Omega) \times L^2(\Omega)$, then the fine scale-norm $\|\cdot\|_{\mathbf{K}^{-1}}$ is given by:

$$(22) \quad \|\mathbf{w}\|_{\mathbf{K}^{-1}}^2 = \int_{\Omega} (\mathbf{w}, \mathbf{K}^{-1} \mathbf{w}) dx$$

Where (\cdot, \cdot) is the Euclidean inner product on R^2 . This is an L^2 norm with the weight function \mathbf{K}^{-1} . If we choose \mathbf{K} as the weight function we get the energy norm, which is consistent with (22):

$$(23) \quad \|w\|_{\mathbf{K}}^2 = A(w, w) = \int_{\Omega} \nabla w \cdot \mathbf{K} \nabla w dx$$

These norms will also appear in a mixed finite element formulation.

The following can be proved [18, 26]:

5.1. A posteriori error estimate. Let \mathbf{K} and \mathbf{K}^0 be symmetrical and positive definite tensors. Assume that \mathbf{K}^0 is the result of an upscaling technique applied to \mathbf{K} , such that:

$$(24) \quad \int_{\Omega} [\mathbf{K}^0 \nabla p^0(\mathbf{K}^0), \nabla p^0(\mathbf{K}^0)] dx \leq \int_{\Omega} [\mathbf{K} \nabla p, \nabla p] dx$$

and

$$(25) \quad \|\mathbf{K}\|_{L^\infty} \leq M \text{ and } \|\mathbf{K}^0\|_{L^\infty} \leq m \leq M$$

Let p^0 and p be solutions of respectively eq.(20) and eq. (19), and let \mathbf{v} and \mathbf{v}^0 be the velocities given by $\mathbf{v} = -\mathbf{K} \nabla p$ and $\mathbf{v}^0 = -\mathbf{K}^0 \nabla p^0$. We have the following error estimate:

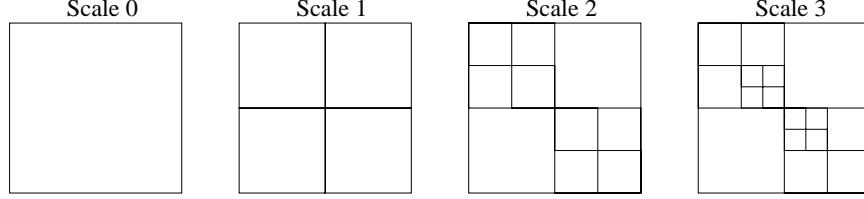


FIGURE 4. The figure shows a nested sequence of subdomains,
 $\Omega_i \subset \Omega$.

$$(26) \quad \|\mathbf{v} - \mathbf{v}^0\|_{\mathbf{K}^{-1}} \leq \|\mathbf{I}_0 \mathbf{v}^0\|_{\mathbf{K}^{-1}}$$

and

$$(27) \quad \|p - p^0\|_{\mathbf{K}} \leq \|\mathbf{I}_0 \nabla p^0\|_{\mathbf{K}}$$

where $(\mathbf{I}_0)^2 = \mathbf{I} - \mathbf{K}^{-1}\mathbf{K}^0$

The assumption (24) states that the dissipation in the problem is constant or decreasing during upscaling. Conservation of dissipation is one of the criteria used for judging the quality of an upscaling procedure see Section 4. The permeability representation given in Section 2 is consistent with the assumptions given in the theorem.

The a posteriori estimate of the modeling error is built on the results in [26]. It is proved [26] that the upscaling based on this norm gives very good results and as noted, the estimate does not involve the fine scale solution. The error estimate gives the basis for an adaptive upscaling technique. Together with domain decomposition, it gives a strong modeling procedure. The following algorithm is consistent with the hierarchical permeability representation given in Section 2:

Algorithm for an Adaptive Domain Decomposition Solution

Solve on scale 0. (Global coarse grid solve) This gives P_Ω^0 and \mathbf{v}_Ω^0
 While(*error* \geq *tol* and *scale* \leq *maxscale*)

Solve on all active local domains using the MRA representation.
 (After this step our data structure contains only inactive domains.)
 for(*i* = 1; *i* \leq *number_of_domains_on_this_scale*; *i*++)
 if(*local_error* \geq *tol*)
 Generate four new active subdomains of this local domain.
 end if
 if(*local_error* \geq *local_error_prev*)
 local_error_prev = *local_error*
 end if
 end for
 error = *local_error_prev*
 end while(...)

The separation of the domains into two classes, active and non-active, ensures that we do not waste time recomputing on domains which are not split up as we move up in the hierarchy. Thus the iteration will just involve these four sub domains

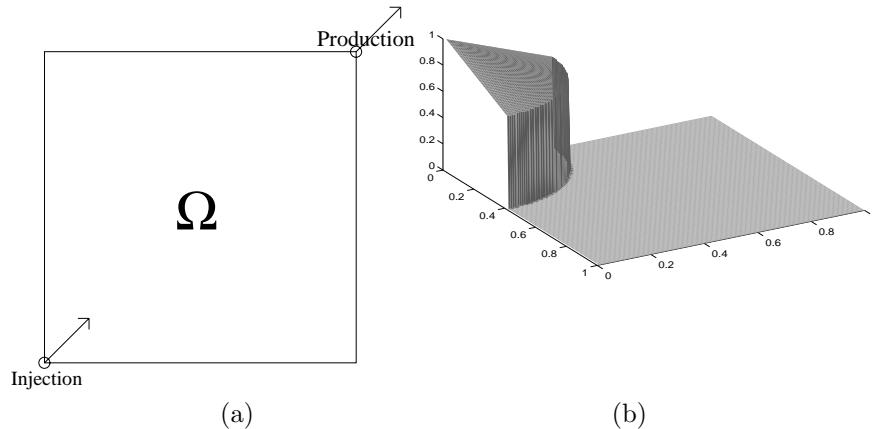


FIGURE 5. Experimental setting (a) and initial saturation profile (b).

if only one domain is refined. On the other hand if two neighboring domains are refined, the iteration process will involve eight domains. Figure 4 shows a situation where all scales from scale 1 to scale 3 are present in the solution.

6. Numerical experiments

In this section we will present numerical experiments to demonstrate the effect of the error estimate and the ‘‘high-cut’’ upscaling technique. The computations will be based on the permeability data given by Figure 3. All experiments are based on the ‘‘quarter of a fivespot’’ problem. Our computational domain will be the unit square. Figure 5 shows a typical configuration along with the initial saturation profile.

6.1. Reference Solution. The main objective of the experiment is to better understand the behavior of the error estimate in theorem (5.1) together with domain decomposition methods, based on local solvers [18]. We define the effectivity indices, η_K , [28], measuring the quality of the error prediction as:

$$(28) \quad \eta_K = \frac{\|I^0 v^0\|_{K^{-1}}^2}{\|v^0 - v\|_{K^{-1}}^2}$$

To create a reference solution, we apply a global solver and identical grid on all scales. This grid has 128×128 cells and corresponds to scale 7. This means that on scale 7 each cell is assigned a permeability value and we use scale 7 as our reference scale. Figure 6 show the velocity and saturation fields for scale 7 and 4. The results are based on the permeability field given by Figure 3. The most apparent feature is the complexity in the fields. Both absolute value and direction changes radically over short distances. The saturation are shown at $t = 0.14$. We see that the main flowpattern are the same for the models, but as expected, some of the small scale variation is lost on scale 4 [24].

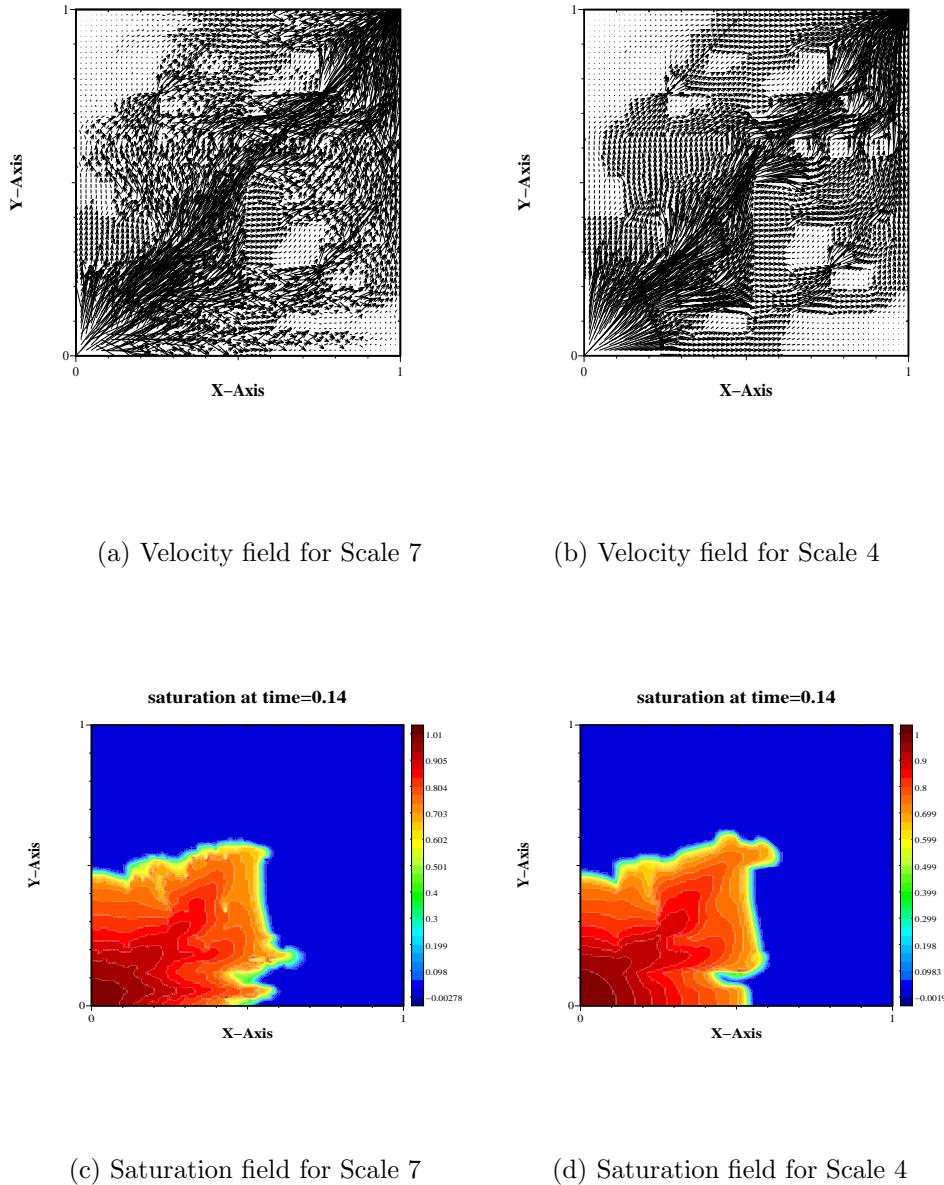


FIGURE 6. The figure shows the velocity and saturation fields for a highcut upscaling with global solvers. Figure (a) and (c) represent the reference solution. Scale 7 → 128×128 cells and scale 4 → 16×16 cells.

TABLE 2. The effectivity indices for η_K for scale 4 - 6

Scale	η_K
6	1.45
5	1.48
4	1.54

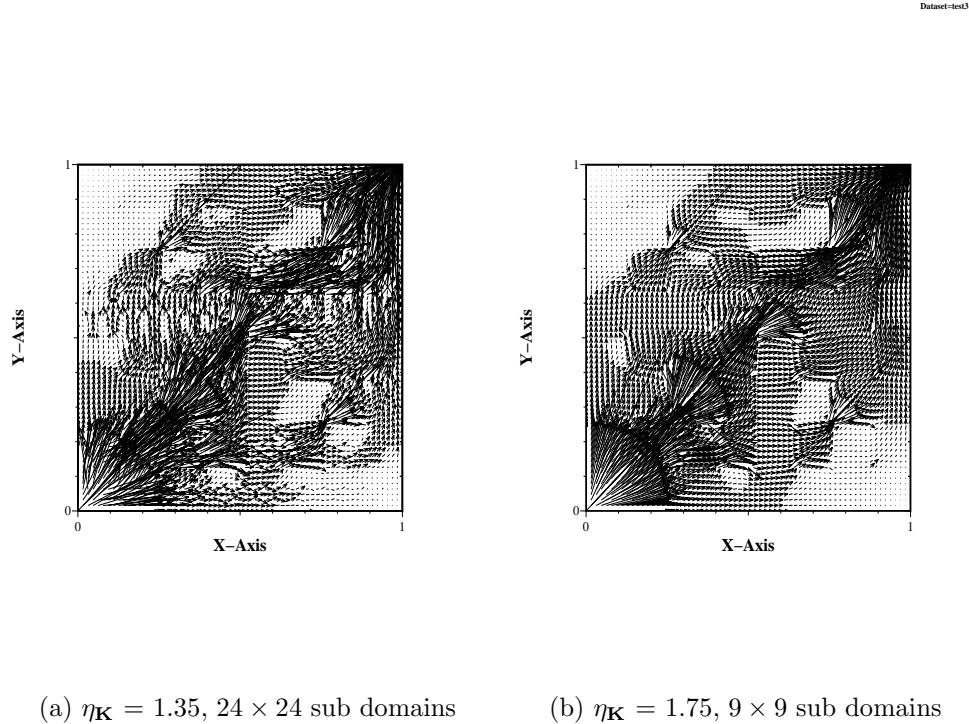


FIGURE 7. The figure shows the velocity field for a highcut upscaling technique coupled with localized solvers through the adaptive upscaling algorithm. The results should be compared with the reference solution (Scale 7) given in Figure 6

Table 2 shows the effectivity indices for each scale, which shows that η_K vary as expected.

It can be shown that dissipation is fairly well conserved in this case [18].

6.2. Local solvers. In this section we will test the local adaptive upscaling algorithm (5.1) together with local solvers [18]. To capture the global communication in the problem we had to apply coarse grid solvers at each scale. This was apparent only when the velocity field showed radical changes in both absolute value and direction. For permeability fields where these changes were less rapid a coarse grid solver was no longer necessary [18]. The domain decomposition has been based on the multiplicative Schwarz algorithm. Figure 7 shows the velocity field for an upscaling with the adaptive algorithm (5.1). The plot shows the velocity field corresponding for two values of the error estimate. Note that even when the

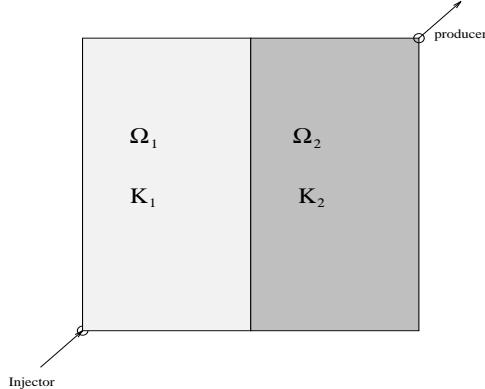


FIGURE 8. The computational domain Ω .

quality of the error estimation is poor, $\eta_K = 1.75$, most of the global transport is still captured using only 9×9 local domains. The results in Figure 7 should be compared to the reference velocity field given in Figure 6. As the quality of the error estimation improves the solution is almost identical to the reference solutions [18] and even for $\eta_K = 1.35$, 24×24 sub domains, the velocity field is very accurate. The iteration error was chosen very small such that the iteration error did not influence on the estimate of the modeling error. The two solutions will, however, never be identical due to large areas where the mean velocities are small such that the mesh is not refined. The differences are not visible on the plots and have only a negligible effect on the global flow.

In most cases, the local coarsening of meshes by upscaling, will reduce the computational cost substantially. With the adaptive technique proposed, the modeling error can still be kept under control.

7. Two sediment model

The solution procedure described in Section 3 can be extended to models consisting of several types of sediments. We will limit the presentation to a two sediment model.

We consider a rectangular domain Ω which has an interior boundary Γ , at $x = 0.5$, separating the two sediments. We assume that both Ω_1 and Ω_2 are homogeneous. All the parameters like absolute and relative permeability and capillary pressure, may be discontinuous across such interfaces [36, 19, 12, 13, 14, 20]. We will assume that only the absolute permeability is discontinuous across Γ , which means that also the capillary pressure (4) is discontinuous. In order to satisfy the interface relations (5) and (7), the saturation S also has to be discontinuous across Γ .

This discontinuity is handled by introducing discontinuous trial functions [12, 14] in the weak formulation of the saturation equation (10). The jump discontinuity in the trialfunctions [12, 14] are determined from the interface relations (5) and (7). A problem with an injector well in lower left corner and a production well in upper right corner, as shown in Figure 8 is studied. The numerical results obtained with the adaptive local grid refinement at the front and a fixed local refinement at the interior boundary in [12, 34, 13] are compared

TABLE 3. Time and well data for the test cases.

Parameter	value	Description
dt	0.001	time step
t	[0,48]	time intervall
inj_rate	0.2	injection rate
prod_rate	-0.2	production rate

TABLE 4. The permeability fields defining test cases a and b.

case:	$K(\Omega_1)$	$K(\Omega_2)$
a	1.0	0.1
b	0.1	1.0

with equivalent results computed on a uniform global mesh. The local refinement is obtained by domain decomposition methods described in Section 3. The agreement between the two approaches is very good.

We choose the same type of initial condition for the saturation as given in Figure 5. This represents an established front, located away from the injection well.

Saturation solutions will be shown at four time levels, $t = 0.24$, and $t = 0.48$. Input data are given in Table 3 and Table 4, while the injection and production rates are given in Table 3 together with the time parameters.

Figure 9 gives computed saturations for the test cases a and b in Table 4. In Figure 9 (a) and (c) (Testcase a) water is injected in a high permeability sediment and oil is produced in a low permeable sediment. In Figure 9 (b) and (d) (Testcase b) water is injected in a low permeable sediment, and oil is produced in a high permeable sediment.

The total Darcy velocity and the capillary diffusion depend on the permeability field \mathbf{K} . The effects of different permeabilities and capillary forces in the two regions Ω_1 and Ω_2 combined with the effect of the boundary conditions (5), (7), are clearly seen in Figure 9 (a) - (d). The saturation jump at the internal boundary is positive for testcase a and negative for testcase b [12, 14].

We observe that the saturation front is more smeared in the high permeable region than in the low permeable region, consistent with the magnitude of the diffusion term as given in Figure 1.

As noted, the results shown are computed both with uniform grids and with local refinement based on domain decomposition [12]. The grid is adaptively refined at the saturation fronts [8, 12] and at the internal boundary Γ . Except for small interpolation effects at the boundary between the coarse and the refined meshes, the results are similar.

The results are extended to models where both absolute and relative permeability and capillary pressure are given by different functional form in the two sediments [12, 14]. Work is under way to extend the models to heterogeneous multisediment models and models with fractures. The local refinement of meshes at interfaces and saturation fronts or the local coarsening of meshes by upscaling, will reduce the computational cost substantially.

Mon Apr 1 15:31:08 1996

1

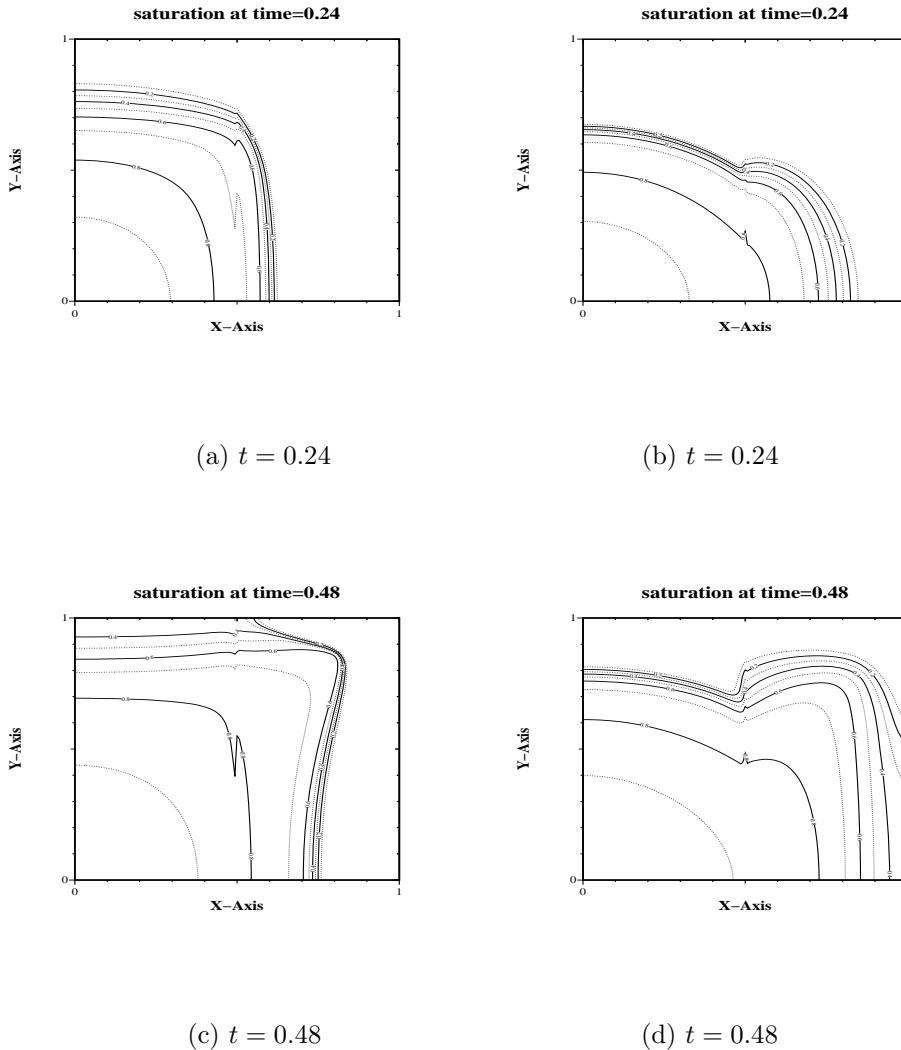


FIGURE 9. Saturation profile for five spot problem at time levels $t = 0.24$, and $t = 0.48$. $K(\Omega_1) = 1$ and $K(\Omega_2) = 0.1$ in Figure (a) and (c). $K(\Omega_1) = 0.1$ and $K(\Omega_2) = 1$ in Figure (b) and (d).

8. Conclusion

In this work we have focused on solution techniques for two phase porous media flow, where the models may be heterogeneous or consist of different types of sediments. An adaptive upscaling technique is given. This gives a flexible and powerful procedure for local grid refinement based on domain decomposition. Work is under way to extend the methods to more complex models like flow in fractured rock.

References

1. B. Amaziane, A Bourgeat, and J. V. Koebe, *Numerical simulation and homogenization of diphasic flow in heterogeneous reservoirs*, Proceedings II European conf. on Math. in Oil Recovery (Arle, France) (O. Guillon D. Guerillot, ed.), 1990.
2. O. Axelsson and V.A. Barker, *Finite element solution of boundary value problems*, Academic Press, London, 1984.
3. J.W. Barrett and K.W. Morton, *Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems*, Computer Methods in Applied Mechanics and Engineering **45** (1984), 97–122.
4. Ø. Bøe, *Finite-element methods for porous media flow*, Ph.D. thesis, Department of Applied Mathematics, University of Bergen, Norway, 1990.
5. G. Chavent and J.Jaffre, *Mathematical models and finite elements for reservoir simulation*, North-Holland, Amsterdam, 1987.
6. C. K. Chu, *An introduction to wavelets*, Academic Press, Boston, 1992.
7. G. Dagan, *Flow and transport in porous formations*, Springer-Verlag, Berlin-Heidelberg, 1989.
8. H.K. Dahle, M.S. Espedal, and O. Sævareid, *Characteristic, local grid refinement techniques for reservoir flow problems*, Int. Journal for Numerical Methods in Engineering **34** (1992), 1051–1069.
9. I. Daubechies, *Ten lectures on wavelets*, Conf. Ser.Appl. Math. SIAM, Philadelphia, 1992.
10. J. Douglas, Jr. and T.F. Russell, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM Journal on Numerical Analysis **19** (1982), 871–885.
11. L. J. Durlofsky and E. Y. Chang, *Effective permeability of heterogeneous reservoir regions*, Proceedings II European conf. on Math. in Oil Recovery (Arle, France) (O. Guillon D. Guerillot, ed.), 1990.
12. B. G. Ersland, *On numerical methods for including the effect of capillary pressure forces on two-phase, immiscible flow in a layered porous media*, Ph.D. thesis, Department of Applied Mathematics, University of Bergen, Norway, 1996.
13. B. G. Ersland and M. S. Espedal, *Domain decomposition method for heterogeneous reservoir flow*, Proceedings Eight International Conference on Domain Decomposition Methods, Beijing, 1995 (London) (R. Glowinski, J. Periaux, Z-C. Shi, and O. Widlund, eds.), J.Wiley, 1997.
14. M. S. Espedal, B. G. Ersland, K. Hersvik, and R. Nybø, *Domain decomposition based methods for flow in a porous media with internal boundaries*, Submitted to: Computational Geoscience, 1997.
15. M. S. Espedal and R.E. Ewing, *Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow*, Computer Methods in Applied Mechanics and Engineering **64** (1987), 113–135.
16. J. Frøyen and M. S. Espedal, *A 3D parallel reservoir simulator*, Proceedings V European conf. on Math. in Oil Recovery (Leoben,Austria) (M. Kriebernegg E. Heinemann, ed.), 1996.
17. R. Hansen and M. S. Espedal, *On the numerical solution of nonlinear flow models with gravity*, Int. J. for Num. Meth. in Eng. **38** (1995), 2017–2032.
18. K. Hersvik and M. S. Espedal, *An adaptive upscaling technique based on an a-priori error estimate*, Submitted to: Computational Geoscience, 1997.
19. J. Jaffre, *Flux calculation at the interface between two rock types for two-phase flow in porous media*, Transport in Porous Media **21** (1995), 195–207.
20. J. Jaffre and J. Roberts, *Flux calculation at the interface between two rock types for two-phase flow in porous media*, Proceedings Tenth International Conference on Domain Decomposition Methods, Boulder, (1997).
21. K. Hvistendahl Karlsen, K. Brusdal, H. K. Dahle, S. Evje, and K. A. Lie, *The corrected operator splitting approach applied to nonlinear advection-diffusion problem*, Computer Methods in Applied Mechanics and Engineering (1998), To appear.
22. P. R. King, *The use of renormalization for calculating effective permeability*, Transport in Porous Media **4** (1989), 37–58.
23. T. F. M. Kortekaas, *Water/oil displacement characteristics in crossbedded reservoir zones*, Society of Petroleum Engineers Journal (1985), 917–926.
24. P. Langlo and M. S. Espedal, *Macrodispersion for two-phase immiscible flow in porous media*, Advances in Water Resources **17** (1994), 297–316.

25. K. W. Morton, *Numerical solution of convection-diffusion problems*, Applied Mathematics and Mathematical Computation, 12, Chapman & Hall, London, 1996.
26. B. F. Nielsen and A. Tveito, *An upscaling method for one-phase flow in heterogeneous reservoirs; a weighted output least squares approach*, Preprint, Dep. of informatics, Univ. of Oslo, Norway, 1996.
27. S. Nilsen and M. S. Espedal, *Upscaling based on piecewise bilinear approximation of the permeability field*, Transport in Porous Media **23** (1996), 125–134.
28. J. T. Oden, T. Zohdi, and J. R. Cho, *Hierarchical modelling, a posteriori error estimate and adaptive methods in computational mechanics*, Proceedings Comp. Meth. in Applied Sciences (London) (J. A. Desideri et.al., ed.), J. Wiley, 1996, pp. 37–47.
29. R. Rannacher and G. H. Zhou., *Analysis of a domain splitting method for nonstationary convection-diffusion problems*, East-West J. Numer. math. **2** (1994), 151–174.
30. Y. Rubin, *Stochastic modeling of macrodispersion in heterogeneous porous media*, Water Resources Res. **26** (1991), 133–141.
31. O. Sævareid, *On local grid refinement techniques for reservoir flow problems*, Ph.D. thesis, Department of Applied Mathematics, University of Bergen, Norway, 1990.
32. B. Smith, P. Bjørstad, and W. Gropp, *Domain decomposition*, Cambridge University Press, Cambridge, 1996.
33. X. C. Tai and M. S. Espedal, *Space decomposition methods and application to linear and nonlinear elliptic problems*, Int. J. for Num. Meth. in Eng. (1998), To appear.
34. X. C. Tai, O. W. Johansen, H. K. Dahle, and M. S. Espedal, *A characteristic domain splitting method for time-dependent convection diffusion problems*, Proceedings Eight International Conference on Domain Decomposition Methods, Beijing, 1995 (London) (R. Glowinski, J. Periaux, Z-C. Shi, and O. Widlund, eds.), J. Wiley, 1997.
35. T.Bu and L.B. Håøy, *On the importance of correct inclusion of capillary pressure in reservoir simulation*, Proceedings European IOR - Symposium, 1995.
36. C. J. van Duijn, J. Molenaar, and M. J. de Neef, *The effect of the capillary forces on immiscible two-phase flow in heterogeneous porous media*, Transport in Porous Media **21** (1995), 71–93.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BERGEN, N-5008 BERGEN, NORWAY
E-mail address: Magne.Espedal@mi.uib.no

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BERGEN, N-5008 BERGEN, NORWAY
E-mail address: Karl.Hersvik@mi.uib.no

STATOIL A.S., N-5020 BERGEN, NORWAY
E-mail address: BGE@statoil.no

A Fictitious Domain Method with Distributed Lagrange Multipliers for the Numerical Simulation of Particulate Flow

Roland Glowinski, Tsorng-Whay Pan, Todd I. Hesla, Daniel D. Joseph,
and Jacques Periaux

1. Introduction

The main goal of this article, which generalizes [4] considerably, is to discuss the *numerical simulation of particulate flow* for mixtures of *incompressible viscous fluids* and *rigid particles*. Such flow occurs in liquid/solid *fluidized beds*, *sedimentation*, and other applications in Science and Engineering. Assuming that the number of particles is sufficiently large, those simulations are useful to adjust parameters in the *homogenized models* approximating the above *two-phase flow*.

From a *computational* point of view, the methodology to be discussed in this article combines *distributed Lagrange multipliers* based *fictitious domain methods*, which allow the use of *fixed structured finite element grids* for the fluid flow computations, with *time discretizations* by *operator splitting à la Marchuk-Yanenko* to decouple the various computational difficulties associated to the simulation; these difficulties include *collisions* between particles, which are treated by *penalty* type methods. After validating the numerical methodology discussed here by comparison with some well documented *two particle - fluid flow* interactions, we shall present the results of two and three dimensional particulate flow simulations, with the number of particles in the range $10^2 - 10^3$; these results include the simulation of a *Rayleigh-Taylor instability* occurring when a sufficiently large number of particles, initially at rest, are positioned regularly over a fluid of smaller density, in the presence of gravity.

The methods described in this article will be discussed with more details (of computational and physical natures) in [6]. Actually, ref. [6] will contain, also, many references to the work of several investigators, showing that the most popular methodology to simulate particulate flow has been so far the one based on *ALE (Arbitrary Lagrange-Euler)* techniques; these methods are clearly more complicated to implement than those described in this article (particularly on *parallel* platforms).

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 65M60, 70F99, 76P05.

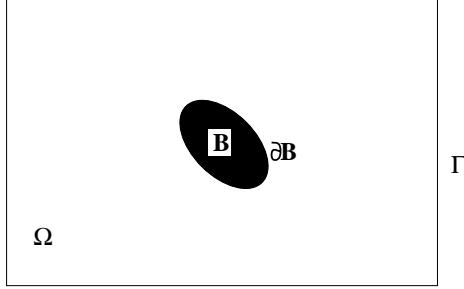


FIGURE 1. The rigid body B and the flow region $\Omega \setminus \bar{B}$

2. A model problem

For simplicity, we shall consider first the motion of a *unique rigid body* B , surrounded by a *Newtonian incompressible viscous fluid*. From a geometrical point of view, the situation is the one depicted in Figure 1.

The rigid body $B(t)(= B)$ is contained in a region $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$, in practice). The *fluid flow* is modelled by the following *Navier-Stokes equations* (with obvious and/or classical notation):

- (1) $\rho_f \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = \rho_f \mathbf{g} + \nabla \cdot \boldsymbol{\sigma} \text{ in } \Omega \setminus \overline{B(t)},$
- (2) $\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \setminus \overline{B(t)},$
- (3) $\mathbf{u}(x, 0) = \mathbf{u}_0(x), x \in \Omega \setminus \overline{B(0)}, \text{ with } \nabla \cdot \mathbf{u}_0 = 0,$
- (4) $\mathbf{u} = \mathbf{g}_0 \text{ on } \Gamma.$

We remind that for *Newtonian fluids* the *stress tensor* $\boldsymbol{\sigma}$ is defined by

$$(5) \quad \boldsymbol{\sigma} = -p\mathbf{I} + \nu_f(\nabla \mathbf{u} + \nabla \mathbf{u}^t).$$

Assuming that a *no-slip condition* holds on $\partial B(t)$, the *rigid body motion* of $B(t)$, combined with the *incompressibility condition* (2), implies that $\int_{\Gamma} \mathbf{g}_0 \cdot \mathbf{n} d\Gamma = 0$.

For further simplicity, we shall assume that $\Omega \subset \mathbb{R}^2$, but there is no basic difficulty to generalize the following considerations to three-dimensional particulate flow. Denoting by \mathbf{V} (resp., ω) the *velocity of the center of mass* G (resp., the *angular velocity*) of the rigid body B , we have for the motion of B the following *Newton's equations*:

$$(6) \quad M\dot{\mathbf{V}} = \mathbf{F} + Mg,$$

$$(7) \quad I\dot{\omega} = T,$$

$$(8) \quad \dot{G} = \mathbf{V},$$

with the *force* \mathbf{F} and *torque* T , resulting from the *fluid-particle interaction*, given by

$$(9) \quad \mathbf{F} = \int_{\partial B} \boldsymbol{\sigma} \mathbf{n} d\gamma,$$

$$(10) \quad T = \int_{\partial B} (\vec{G} \times \boldsymbol{\sigma} \mathbf{n}) \cdot \mathbf{e}_3 d\gamma,$$

where, in (10), $\mathbf{e}_3 = \{0, 0, 1\}$ if we assume that Ω is contained in the plane x_1Ox_2 . The no-slip boundary condition mentioned above implies that on ∂B we have

$$(11) \quad \mathbf{u}(x, t) = \mathbf{V}(t) + \boldsymbol{\omega}(t) \times \vec{G}x, \quad \forall x \in \partial B(t),$$

with $\boldsymbol{\omega} = \{0, 0, \omega\}$. Of course, I is the *moment of inertia* of B , with respect to G .

3. A global variational formulation

We introduce first the following *functional spaces*

$$\begin{aligned} V_{g_0(t)} &= \{\mathbf{v} | \mathbf{v} \in H^1(\Omega \setminus \overline{B(t)})^2, \mathbf{v} = \mathbf{g}_0(t) \text{ on } \Gamma\}, \\ V_0(t) &= \{\mathbf{v} | \mathbf{v} \in H^1(\Omega \setminus \overline{B(t)})^2, \mathbf{v} = 0 \text{ on } \Gamma, \mathbf{v} = \mathbf{Y} + \boldsymbol{\theta} \times \vec{G}x \\ &\quad \text{on } \partial B, \mathbf{Y} \in \mathbb{R}^2, \boldsymbol{\theta} \in \mathbb{R}\}, \\ L_0^2(\Omega \setminus \overline{B(t)}) &= \{q | q \in L^2(\Omega \setminus \overline{B(t)}), \int_{\Omega \setminus \overline{B(t)}} q dx = 0\} \end{aligned}$$

with $\boldsymbol{\theta} = \{0, 0, \theta\}$. By application of the *virtual power principle* (ref. [7]) we are led to look for:

$$\mathbf{u}(t) \in V_{g_0(t)}, p \in L_0^2(\Omega \setminus \overline{B(t)}), \mathbf{V}(t) \in \mathbb{R}^2, \omega(t) \in \mathbb{R} \text{ such that}$$

$$(12) \quad \begin{cases} \rho_f \int_{\Omega \setminus \overline{B(t)}} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} dx + \rho_f \int_{\Omega \setminus \overline{B(t)}} (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} dx - \int_{\Omega \setminus \overline{B(t)}} p \nabla \cdot \mathbf{v} dx \\ + 2\nu_f \int_{\Omega \setminus \overline{B(t)}} \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) dx + M(\dot{\mathbf{V}} - \mathbf{g}) \cdot \mathbf{Y} + I\dot{\omega}\theta \\ = \rho_f \int_{\Omega \setminus \overline{B(t)}} \mathbf{g} \cdot \mathbf{v} dx, \quad \forall \mathbf{v} \in V_0(t), \quad \forall \{\mathbf{Y}, \theta\} \in \mathbb{R}^3, \end{cases}$$

$$(13) \quad \int_{\Omega \setminus \overline{B(t)}} q \nabla \cdot \mathbf{u}(t) dx = 0, \quad \forall q \in L^2(\Omega \setminus \overline{B(t)}),$$

$$(14) \quad \mathbf{u} = \mathbf{g}_0 \text{ on } \Gamma,$$

$$(15) \quad \mathbf{u} = \mathbf{V} + \boldsymbol{\omega} \times \vec{G}x \text{ on } \partial B(t),$$

$$(16) \quad \mathbf{u}(x_0) = \mathbf{u}_0(x), \quad \forall x \in \Omega \setminus \overline{B(0)}, \text{ with } \nabla \cdot \mathbf{u}_0 = 0,$$

$$(17) \quad \mathbf{V}(0) = \mathbf{V}_0, \quad \omega(0) = \omega_0.$$

In (12), $dx = dx_1 dx_2$, and the *rate-of-strain tensor* $\mathbf{D}(\mathbf{v})$ is given by

$$(18) \quad \mathbf{D}(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^t),$$

and we have, for $G(t)(= G)$ in (15),

$$(19) \quad G(t) = G_0 + \int_0^t \mathbf{V}(s) ds.$$

4. A fictitious domain formulation

The *fictitious domain method* discussed below, offers an alternative to the *ALE* methods investigated in [9], [14], [13]. The basic idea is quite simple and can be summarized as follows:

- (i) *Fill each particle with the surrounding fluid.*
- (ii) *Impose a rigid body motion to the fluid inside each particle.*

- (iii) Relax the rigid body motion inside each particle by using a distributed Lagrange multiplier defined over the space region occupied by the particles.

In the following, we shall assume that B is made of an homogeneous material of density ρ_s . Starting from the global variational formulation (12)-(17) and following steps (i) to (iii) leads to the following *generalized variational problem*, where $\lambda(t)$ is the *distributed Lagrange multiplier* forcing at time t rigid body motion for the fluid "filling" body B :

$$\text{Find } \mathbf{U}(t) \in W_{g_0(t)} = \{\mathbf{v} | \mathbf{v} \in H^1(\Omega)^2, \mathbf{v} = \mathbf{g}_0(t) \text{ on } \Gamma\},$$

$$P(t) \in L_0^2(\Omega) = \{q | q \in L^2(\Omega), \int_{\Omega} q dx = 0\},$$

$$\lambda(t) \in \Lambda(t) = L^2(B(t))^2 \text{ or } \lambda(t) = H^1(B(t))^2, \text{ so that}$$

$$(20) \quad \begin{cases} \rho_f \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} dx + \rho_f \int_{\Omega} (\mathbf{U} \cdot \nabla) \mathbf{U} \cdot \mathbf{v} dx - \int_{\Omega} P \nabla \cdot \mathbf{v} dx \\ + 2\nu_f \int_{\Omega} \mathbf{D}(\mathbf{U}) : \mathbf{D}(\mathbf{v}) dx + (1 - \rho_f / \rho_s) M(\dot{\mathbf{V}} - \mathbf{g}) \cdot \mathbf{Y} \\ + (1 - \rho_f / \rho_s) I \omega \theta - \langle \lambda, \mathbf{v} - \mathbf{Y} - \boldsymbol{\theta} \times \vec{G}x \rangle_{B(t)} = \rho_f \int_{\Omega} \mathbf{g} \cdot \mathbf{v} dx, \\ \forall \mathbf{v} \in H_0^1(\Omega)^2, \forall \{\mathbf{Y}, \theta\} \in \mathbb{R}^3, \end{cases}$$

$$(21) \quad \int_{\Omega} q \nabla \cdot \mathbf{U} dx = 0, \forall q \in L^2(\Omega),$$

$$(22) \quad \langle \mu, \mathbf{U} - \mathbf{V} - \boldsymbol{\omega} \times \vec{G}x \rangle_{B(t)} = 0, \forall \mu \in \Lambda(t),$$

$$(23) \quad \mathbf{U} = \mathbf{g}_0 \text{ on } \Gamma,$$

$$(24) \quad \mathbf{U}(x, 0) = \mathbf{U}_0(x), x \in \Omega, \text{ with } \nabla \cdot \mathbf{U}_0 = 0 \text{ and } \mathbf{U}_0|_{\Omega \setminus \overline{B(t)}} = \mathbf{u}_0,$$

$$(25) \quad \mathbf{V}(0) = \mathbf{V}_0, \omega(0) = \omega_0, G(0) = G_0.$$

If (20)-(25) hold, it can be easily shown that $\mathbf{U}(t)|_{\Omega \setminus \overline{B(t)}} = \mathbf{u}(t)$, $P(t)|_{\Omega \setminus \overline{B(t)}} = p(t)$, where $\{\mathbf{u}(t), p(t)\}$ completed by $\{\mathbf{V}(t), \omega(t)\}$ is a solution of the global variational problem (12)-(17). The above formulation deserves several remarks; we shall limit ourselves to

REMARK 1. From a *mathematical* point of view, the good choice for $\Lambda(t)$ is $H^1(B(t))^2$ with $\langle \cdot, \cdot \rangle_{B(t)}$ defined by either

$$(26) \quad \langle \mu, \mathbf{v} \rangle_{B(t)} = \int_{B(t)} (\mu \cdot \mathbf{v} + d^2 \nabla \mu : \nabla \mathbf{v}) dx,$$

or

$$(27) \quad \langle \mu, \mathbf{v} \rangle_{B(t)} = \int_{B(t)} (\mu \cdot \mathbf{v} + d^2 \mathbf{D}(\mu) : \mathbf{D}(\mathbf{v})) dx,$$

where, in (26) and (27), d^2 is a *scaling factor*, with d a *characteristic length*, an obvious choice for d being the *diameter* of B . An obvious advantage of $\langle \cdot, \cdot \rangle_{B(t)}$ defined by (27) is that the differential part of it vanishes if \mathbf{v} is a rigid body motion velocity field.

The choice $\Lambda(t) = L^2(B(t))^2$ is not suitable for the continuous problem, since, in general, \mathbf{u} and P do not have enough regularity for λ to be in $L^2(B(t))^2$. On

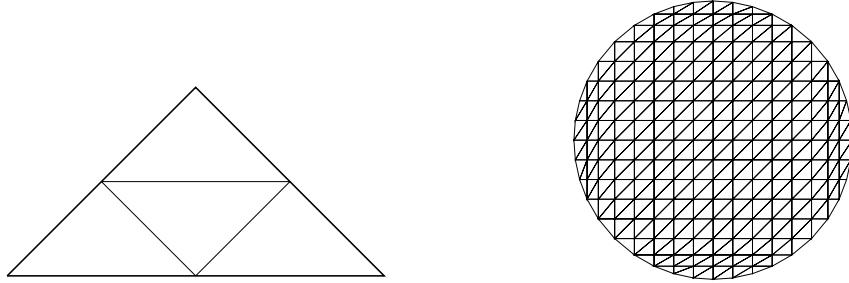


FIGURE 2. Refinement of a triangle of T_{2h}^{Ω} (left) and a triangulation of a disk (right).

the other hand, $\langle \cdot, \cdot \rangle_{B(t)}$ defined by

$$(28) \quad \langle \boldsymbol{\mu}, \mathbf{v} \rangle_{B(t)} = \int_{\Omega} \boldsymbol{\mu} \cdot \mathbf{v} dx$$

makes sense for the *finite dimensional approximations* of problem (20)-(25), in order to force rigid body motion inside B .

REMARK 2. In the case of *Dirichlet boundary conditions* on Γ , and taking the *incompressibility condition* $\nabla \cdot \mathbf{U} = 0$ into account, we can easily show that

$$(29) \quad 2\nu_f \int_{\Omega} \mathbf{D}(\mathbf{U}) : \mathbf{D}(\mathbf{v}) dx = \nu_f \int_{\Omega} \nabla \mathbf{U} : \nabla \mathbf{v} dx, \quad \forall \mathbf{v} \in W_0,$$

which, from a *computational* point of view, leads to a substantial simplification in (20)-(25).

REMARK 3. The distributed Lagrange multiplier approach can be applied to the cases where the particles have different densities and/or shapes. We are currently investigating the extension of this approach to visco-elastic fluid flow.

REMARK 4. The distributed Lagrange multiplier approach discussed here takes full advantage of the particle rigidity. For deformable particles, we can not apply the above method, directly at least.

Further remarks and comments can be found in [6].

5. Finite element approximation of problem (20)-(25)

5.1. Generalities. Concerning the *space approximation* of problem (20)-(25) the main *computational* issues are:

- (i) The approximation of \mathbf{U} and P which are functions defined over Ω .
- (ii) The approximation of the multiplier λ , which is defined over the *moving domain* $B(t)$.
- (iii) The approximation of the *bilinear* functional

$$\{\boldsymbol{\mu}, \mathbf{v}\} \rightarrow \langle \boldsymbol{\mu}, \mathbf{v} \rangle_{B(t)} .$$

The most delicate issue is (iii).

5.2. On the pressure and velocity spaces. In the following, we shall denote by h the pair $\{h_\Omega, h_B\}$, where h_Ω and h_B are space discretization steps associated to *finite element approximations* defined over Ω and B , respectively. Assuming that $\Omega(\subset \mathbb{R}^2)$ is polygonal (as in Figure 1), we introduce a *finite element triangulation* \mathcal{T}_{2h}^Ω of Ω , so that $\cup_{T \in \mathcal{T}_{2h}^\Omega} T = \bar{\Omega}$, and then a twice finer triangulation \mathcal{T}_h^Ω obtained by joining the mid-points of the edges of the triangles of \mathcal{T}_{2h}^Ω , as shown in Figure 2, above.

The *pressure spaces* $L^2(\Omega)$ and $L_0^2(\Omega)$ are approximated by

$$(30) \quad L_h^2 = \{q_h | q_h \in C^0(\bar{\Omega}), q_h|_T \in P_1, \forall T \in \mathcal{T}_{2h}^\Omega\},$$

$$(31) \quad L_{0h}^2 = \{q_h | q_h \in L_h^2, \int_{\Omega} q_h dx = 0\},$$

respectively, with P_1 the space of the polynomials of two variables of degree ≤ 1 . Similarly, we approximate the velocity spaces W_{g_0} and $W_0 (= H_0^1(\Omega)^2)$ by

$$(32) \quad W_{g_0h} = \{\mathbf{v}_h | \mathbf{v}_h \in C^0(\bar{\Omega})^2, \mathbf{v}_h|_T \in P_1^2, \forall T \in \mathcal{T}_h^\Omega, \mathbf{v}_h|_T = \mathbf{g}_{0h}\},$$

$$(33) \quad W_{0h} = \{\mathbf{v}_h | \mathbf{v}_h \in C^0(\bar{\Omega})^2, \mathbf{v}_h|_T \in P_1^2, \forall T \in \mathcal{T}_h^\Omega, \mathbf{v}_h|_T = \mathbf{0}\},$$

respectively; in (32), \mathbf{g}_{0h} is an approximation of \mathbf{g}_0 so that $\int_{\Gamma} \mathbf{g}_{0h} \cdot \mathbf{n} d\Gamma = 0$.

The above pressure and velocity finite element spaces - and their 3-D generalizations - are classical ones, concerning the approximation of the Navier-Stokes equations for incompressible viscous fluids (see, e.g., [3] and the references therein for details.)

5.3. Approximation of the multiplier space $\Lambda(t)$. At time t , we approximate the *multiplier space* $\Lambda(t)$ by

$$(34) \quad \Lambda_h(t) = \{\boldsymbol{\mu}_h | \boldsymbol{\mu}_h \in C^0(\bar{B}_h(t))^2, \boldsymbol{\mu}_h|_T \in P_1^2, \forall T \in \mathcal{T}_h^{B(t)}\},$$

where, in (34), $B_h(t)$ is a *polygonal* approximation of $B(t)$ and $\mathcal{T}_h^{B(t)}$ is a triangulation of $B_h(t)$. If $B(t)$ is a *disk*, we take advantage of its *rotation-invariance* by taking for $\mathcal{T}_h^{B(t)}$ the triangulation obtained by translating $\mathcal{T}_h^{B(0)}$ by the vector $\vec{G_0G(t)}(\mathcal{T}_h^{B(0)})$ can be viewed as a triangulation of reference); such a triangulation $\mathcal{T}_h^{B(t)}$ is shown in Figure 2.

If $B(t)$ is *not* a disk, we shall take for $\mathcal{T}_h^{B(t)}$ a triangulation *rigidly* attached to $B(t)$.

5.4. Approximation of the bilinear functional $\langle \cdot, \cdot \rangle_{B(t)}$. Compatibility conditions between h_Ω and h_B . Suppose that the bilinear functional $\langle \cdot, \cdot \rangle_{B(t)}$ is defined by

$$\langle \boldsymbol{\mu}, \mathbf{v} \rangle_{B(t)} = \int_{B(t)} (a\boldsymbol{\mu} \cdot \mathbf{v} + b\nabla\boldsymbol{\mu} : \nabla\mathbf{v}) dx,$$

with $a > 0$ and $b \geq 0$. In order to avoid solving complicated mesh intersection problems between \mathcal{T}_h^Ω and $\mathcal{T}_h^{B(t)}$ we approximate $\langle \cdot, \cdot \rangle_{B(t)}$ (and, in fact $\langle \cdot, \cdot \rangle_{B_h(t)}$) by

$$(35) \quad \{\boldsymbol{\mu}_h, \mathbf{v}_h\} \rightarrow \int_{B_h(t)} [a\boldsymbol{\mu}_h \cdot (\pi_h \mathbf{v}_h) + b\nabla\boldsymbol{\mu}_h : \nabla(\pi_h \mathbf{v}_h)] dx,$$

where, in (35), π_h is the *piecewise linear interpolation* operator which to \mathbf{v}_h associates the *unique* element of $\Lambda_h(t)$ obtained by interpolating linearly \mathbf{v}_h on the triangles of $T_h^{B(t)}$, from the values it takes at the vertices of the above triangulation.

As can be expected with mixed variational formulations, some compatibility conditions have to be satisfied between the spaces used to approximate $\{\mathbf{U}, P\}$ and $\boldsymbol{\lambda}$ (see, e.g., [2], [15] for generalities on the approximation of mixed variational problems and several applications). Concerning the particular problem discussed here, namely (20)-(25), let us say that:

- (i) condition $h_B \ll h_\Omega$ is good to force accurately rigid body motion on $\overline{B(t)}$.
- (ii) condition $h_\Omega \ll h_B$ is good for stability.

Our numerical experiments show that $h_\Omega \simeq h_B$ seems to be the right compromise.

REMARK 5. We can also use collocation methods to force rigid body motion on $\overline{B(t)}$. This approach (inspired from [1]) has been tested and the corresponding results are reported in [6].

5.5. Finite Element approximation of problem (20)-(25). It follows from previous Sections that a quite natural finite element approximation for the mixed variational problem (20)-(25) is the one defined by

Find $\{\mathbf{U}_h, P_h\} \in W_{g_{0h}(t)} \times L_0^2(\Omega)$, $\{\mathbf{V}(t), \omega(t)\} \in \mathbb{R}^3$, $\boldsymbol{\lambda}_h(t) \in \Lambda_h(t)$ so that

$$(36) \quad \begin{cases} \rho_f \int_{\Omega} \frac{\partial \mathbf{U}_h}{\partial t} \cdot \mathbf{v} dx + \rho_f \int_{\Omega} (\mathbf{U}_h \cdot \nabla) \mathbf{U}_h \cdot \mathbf{v} dx - \int_{\Omega} P_h \nabla \cdot \mathbf{v} dx \\ + \nu_f \int_{\Omega} \nabla \mathbf{U}_h : \nabla \mathbf{v} dx + (1 - \rho_f/\rho_s) M \frac{d\mathbf{V}}{dt} \cdot \mathbf{Y} + (1 - \rho_f/\rho_s) I \frac{d\omega}{dt} \theta \\ - < \boldsymbol{\lambda}_h, \pi_h \mathbf{v} - \mathbf{Y} - \boldsymbol{\theta} \times \vec{G}x >_{B_h(t)} = \rho_f \int_{\Omega} \mathbf{g} \cdot \mathbf{v} dx \\ + (1 - \rho_f/\rho_s) M \mathbf{g} \cdot \mathbf{Y}, \forall \mathbf{v} \in W_{0h}, \forall \{\mathbf{Y}, \theta\} \in \mathbb{R}^3, \text{ a.e., } t > 0, \end{cases}$$

$$(37) \quad \int_{\Omega} q \nabla \cdot \mathbf{U}_h dx = 0, \forall q \in L_h^2,$$

$$(38) \quad < \boldsymbol{\mu}, \pi_h \mathbf{U}_h - \mathbf{V} - \boldsymbol{\omega} \times \vec{G}x >_{B_h(t)} = 0, \forall \boldsymbol{\mu} \in \Lambda_h(t),$$

$$(39) \quad \mathbf{U}_h(0) = \mathbf{U}_{0h} (\mathbf{U}_{0h} \simeq \mathbf{U}_0), \text{ with } \int_{\Omega} q \nabla \cdot \mathbf{U}_{0h} dx = 0, \forall q \in L_h^2,$$

$$(40) \quad \mathbf{V}(0) = \mathbf{V}_0, \omega(0) = \omega_0, G(0) = G_0,$$

with $G(t) = G_0 + \int_0^t \mathbf{V}(s) ds$.

6. Time discretization by operator-splitting

6.1. Generalities. Most modern *Navier-Stokes* solvers are based on *Operator-Splitting schemes*, in order to force $\nabla \cdot \mathbf{u} = 0$ via a *Stokes solver* (like in, e.g., [3]) or a *L^2 -projection method*, (like in, e.g., [16]). This approach applies also to the particulate flow problems discussed here. Indeed, these problems contain three basic computational difficulties, namely:

- (i) The *incompressibility condition* $\nabla \cdot \mathbf{U} = 0$ and the related unknown pressure P .
- (ii) *Advection* and *diffusion* operators.
- (iii) The *rigid body motion* of $B(t)$ and the related *Lagrange multiplier* $\boldsymbol{\lambda}(t)$.

To each of the above difficulties is associated a specific operator; the operators associated to (i) and (iii) are essentially *projection operators*. From an abstract point of view, the problems to be solved (*continuous* or *discrete*) have the following structure:

$$(41) \quad \begin{cases} \frac{d\varphi}{dt} + A_1(\varphi, t) + A_2(\varphi, t) + A_3(\varphi, t) = f, \\ \varphi(0) = \varphi_0. \end{cases}$$

To solve (41) we suggest a *fractional-step à la Marchuk-Yanenko* (see [11] and the references therein); these schemes are *first order accurate* only, but very stable and easy to implement; actually they can be made second order accurate by *symmetrization*. Applying the Marchuk-Yanenko scheme to the initial value problem, we obtain (with $\Delta t(> 0)$ a time discretization step):

$$(42) \quad \varphi^0 = \varphi_0,$$

and for $n \geq 0$, we compute φ^{n+1} from φ^n via

$$(43) \quad \frac{\varphi^{n+j/3} - \varphi^{n+(j-1)/3}}{\Delta t} + A_j(\varphi^{n+j/3}, (n+1)\Delta t) = f_j^{n+1},$$

for $j = 1, 2, 3$, with $\sum_{j=1}^3 f_j^{n+1} = f^{n+1}$.

6.2. Application of the Marchuk-Yanenko scheme to particulate flow. With α, β so that $\alpha + \beta = 1$, $0 \leq \alpha, \beta \leq 1$, we *time-discretize* (36)-(40) as follows (the notation is self-explanatory):

$$(44) \quad \mathbf{U}^0 = \mathbf{U}_{0h}, \mathbf{V}^0, \omega^0, G^0 \text{ are given;}$$

for $n \geq 0$, assuming that $\mathbf{U}^n, \mathbf{V}^n, \omega^n, G^n$ are known, solve

$$(45) \quad \begin{cases} \rho_f \int_{\Omega} \frac{\mathbf{U}^{n+1/3} - \mathbf{U}^n}{\Delta t} \cdot \mathbf{v} dx - \int_{\Omega} P^{n+1/3} \nabla \cdot \mathbf{v} dx = 0, \forall \mathbf{v} \in W_{0h}, \\ \int_{\Omega} q \nabla \cdot \mathbf{U}^{n+1/3} dx = 0, \forall q \in L_h^2; \{\mathbf{U}^{n+1/3}, P^{n+1/3}\} \in W_{g_{0h}}^{n+1} \times L_{0h}^2. \end{cases}$$

Next, compute $\mathbf{U}^{n+2/3}, \mathbf{V}^{n+2/3}, G^{n+2/3}$ via the solution of

$$(46) \quad \begin{cases} \rho_f \int_{\Omega} \frac{\mathbf{U}^{n+2/3} - \mathbf{U}^{n+1/3}}{\Delta t} \cdot \mathbf{v} dx + \alpha \nu_f \int_{\Omega} \nabla \mathbf{U}^{n+2/3} \cdot \nabla \mathbf{v} dx + \\ \rho_f \int_{\Omega} (\mathbf{U}^{n+1/3} \cdot \nabla) \mathbf{U}^{n+2/3} \cdot \mathbf{v} dx = \rho_f \int_{\Omega} \mathbf{g} \cdot \mathbf{v} dx, \\ \forall \mathbf{v} \in W_{0h}; \mathbf{U}^{n+1/3} \in W_{g_{0h}}^{n+1}, \end{cases}$$

and

$$(47) \quad \mathbf{V}^{n+2/3} = \mathbf{V}^n + \mathbf{g} \Delta t, G^{n+2/3} = G^n + (\mathbf{V}^n + \mathbf{V}^{n+2/3}) \Delta t / 2.$$

Finally, compute $\mathbf{U}^{n+1}, \boldsymbol{\lambda}^{n+1}, \mathbf{V}^{n+1}, \omega^{n+1}, G^{n+1}$ via the solution of

$$(48) \quad \begin{cases} \rho_f \int_{\Omega} \frac{\mathbf{U}^{n+1} - \mathbf{U}^{n+2/3}}{\Delta t} \cdot \mathbf{v} dx + \beta \nu_f \int_{\Omega} \nabla \mathbf{U}^{n+1} \cdot \nabla \mathbf{v} dx + \\ (1 - \rho_f/\rho_s) \left(I \frac{\omega^{n+1} - \omega^n}{\Delta t} \theta + M \frac{\mathbf{V}^{n+1} - \mathbf{V}^{n+2/3}}{\Delta t} \cdot \mathbf{Y} \right) = \\ < \boldsymbol{\lambda}^{n+1}, \pi_h \mathbf{v} - \mathbf{Y} - \boldsymbol{\theta} \times \overrightarrow{G^{n+2/3} x} >_{B_h^{n+2/3}}, \forall \mathbf{v} \in W_{0h}, \{\mathbf{Y}, \theta\} \in \mathbb{R}^3, \\ < \boldsymbol{\mu}, \pi_h \mathbf{U}^{n+1} - \mathbf{V}^{n+1} - \boldsymbol{\omega} \times \overrightarrow{G^{n+2/3} x} >_{B_h^{n+2/3}} = 0, \forall \boldsymbol{\mu} \in \Lambda_h^{n+2/3}, \\ \mathbf{U}^{n+1} \in W_{goh}^{n+1}, \boldsymbol{\lambda}^{n+1} \in \Lambda_h^{n+2/3}, \mathbf{V}^{n+1} \in \mathbb{R}^2, \omega^{n+1} \in \mathbb{R}, \end{cases}$$

and

$$(49) \quad G^{n+1} = G^n + (V^n + V^{n+1}) \Delta t / 2.$$

Solving problem (45) is equivalent to computing the $L^2(\Omega)$ -projection of \mathbf{U}^n on the space W_{goh}^{n+1} . This can be done easily using an *Uzawa/conjugate gradient algorithm*, preconditioned by the discrete analogue of $-\nabla^2$ for the *homogeneous Neumann boundary condition*; such an algorithm is described in [16]. Problem (46) is a *discrete advection-diffusion problem*; it can be solved by the methods discussed in [3].

Finally, problem (48) has the following - classical - *saddle-point* structure

$$(50) \quad \begin{cases} Ax + By = b, \\ B^t x = c, \end{cases}$$

with A a *symmetric and positive definite matrix*. Problem (48) can also be solved by an *Uzawa/conjugate gradient* algorithm; such an algorithm is described in [4] and [6].

7. Remarks on the computational treatment of particle collisions

In the above sections, we have consider the particular case of a *single particle* moving in a region Ω filled with a Newtonian incompressible viscous fluid; we have implicitly discarded possible boundary/particle collisions. The above methodology can be generalized fairly easily to *many particles* cases, with, however, a computational difficulty: one has to prevent particle interpenetration or particle/boundary penetration. To achieve those goals we have included in the *Newton's equations* (6)-(8) modeling particle motions a *short range repulsive force*. If we consider the particular case of *circular* particles (in 2-D) or *spherical* particles in (3-D), and if P_i and P_j are such two particles, with radiiuses R_i and R_j and centers of mass G_i and G_j , we shall require the *repulsion force* \vec{F}_{ij} between P_i and P_j to satisfy the following properties:

- (i) To be parallel to $\overrightarrow{G_i G_j}$.
- (ii) To verify

$$\begin{cases} |\vec{F}_{ij}| = 0 \text{ if } d_{ij} \geq R_i + R_j + \rho, \\ |\vec{F}_{ij}| = c/\varepsilon \text{ if } d_{ij} = R_i + R_j, \end{cases}$$

with $d_{ij} = |\overrightarrow{G_i G_j}|$, c a scaling factor and ε a "small" positive number.

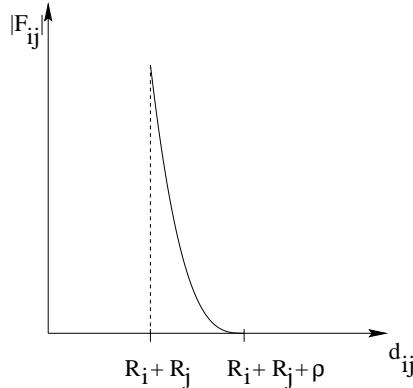


FIGURE 3. Repulsion force behavior

(iii) $|\vec{F}_{ij}|$ has to behave as in Figure 3 for

$$R_i + R_j \leq d_{ij} \leq R_i + R_j + \rho.$$

Parameter ρ is the *range* of the repulsion force; for the simulations discussed in the following Section we have taken $\rho \simeq h_\Omega$.

Boundary/particle collisions can be treated in a similar way (see [6] for details).

REMARK 6. The above collision model is fairly simple and is inspired from *penalty techniques* classically used for the computational treatment of some *contact problems* in Mechanics (see, e.g., [10], [5] for details and applications). Despite its simplicity, this model produces good results, the main reason for that being, in our opinion, that if the fluid is sufficiently viscous and if the fluid and particle densities are close, the collisions - if they occur - are *non violent* ones, implying that the particles which are going to collide move at almost the same velocity just before collision. For more sophisticated models allowing more violent collisions see, e.g., [12] and the references therein.

8. Numerical experiments

8.1. 2 particles case. In order to validate the methodology described in the previous sections, we are going to consider a well-documented case, namely the simulation of the motion of 2 circular particles sedimenting in a two-dimensional channel. We shall apply algorithm (44)-(49) with different mesh sizes and time steps. The computational domain is a finite portion of a channel, which is moving along with the particles. Its x and y dimensions are 2 and 5, respectively. The diameter d of the particles is 0.25. The fluid and particle densities are $\rho_f = 1.0$ and $\rho_s = 1.01$, respectively, and the fluid viscosity is $\nu_f = 0.01$. The initial positions of the two circular particles are at the centerline of the channel with distance 0.5 apart. Initial velocity and angular speed of particles are zero. We suppose that at $t = 0$ the flow is at rest.

For numerical simulations, we have chosen two time steps, $\Delta t = 0.0005$ and 0.00025 , and two mesh sizes for the velocity field, $h_v = 1/192$ and $1/256$. The mesh size for pressure is always $h_p = 2h_v$. The force range, ρ , in which the short range repulsion force is active is $1.5h_v$. For the (stiffness) parameter, ϵ , mentioned in

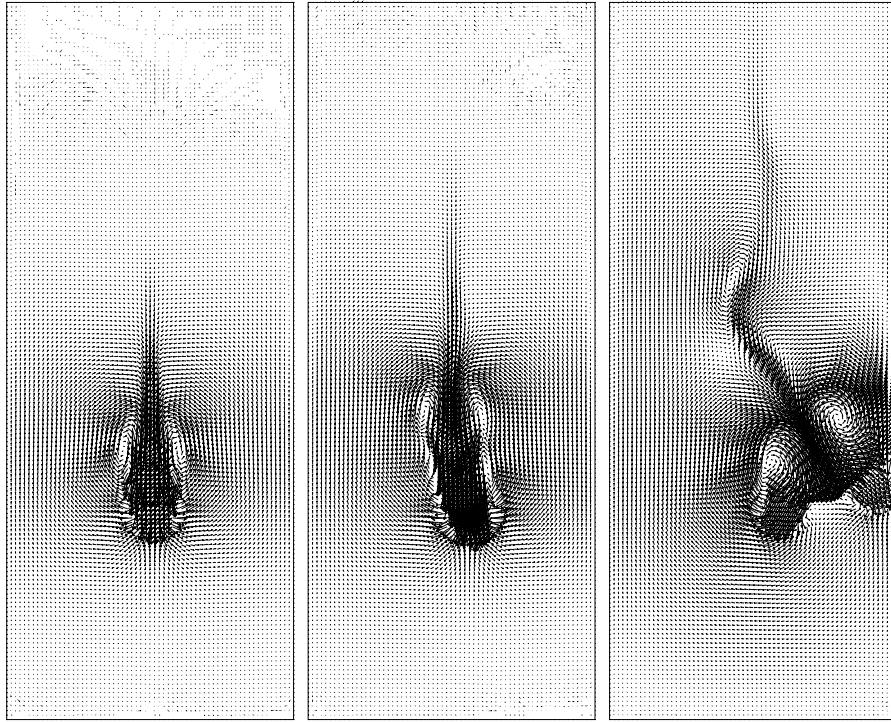


FIGURE 4. Particle position at $t = 0.15, 0.2, 0.3$ (from left to right).

previous Section, we have taken $\epsilon_p = 10^{-5}$ for particle-particle repulsion force and $\epsilon_w = \epsilon_p/2$ for particle-wall repulsion force.

In Figure 4, we can see the fundamental features of two fluidizing particles, i.e., drafting, kissing and tumbling obtained with mesh size $h_v = 1/256$ and time step $\Delta t = 0.0005$. In Figures 5–7, the center of the particles, translation velocity of the center of the particles, and the angular speed of the particles are shown for the cases where the time step is the same, $\Delta t = 0.0005$, and the mesh sizes are $h_v = 1/192$ and $1/256$. The maximal Reynolds numbers in the numerical simulations is about 450. The time at which the two particles are the closest is $t = 0.1665$ in the above two cases. Actually we have a very good agreement between these two simulations until kissing. After kissing, despite the stability breaking which is clearly the manifestation of some instability phenomenon, the simulated particle motions are still very close taking into consideration the difficulty of the problem.

Also in Figures 8–10, similar history graphs are shown which are obtained from the same mesh size, $h_v = 1/192$, and two time steps, $\Delta t = 0.0005$ and 0.00025 . When the time step is $\Delta t = 0.00025$, the maximal Reynolds number in the numerical simulation is about 465 and the time of the smallest distance occurrence is at $t = 0.17125$. We can also find a very good agreement between these two cases until the kissing occurrence.

These results compare qualitatively well with those of Hu, Joseph, and Crochet in [8], which were obtained with different physical parameters and a different numerical methodology.

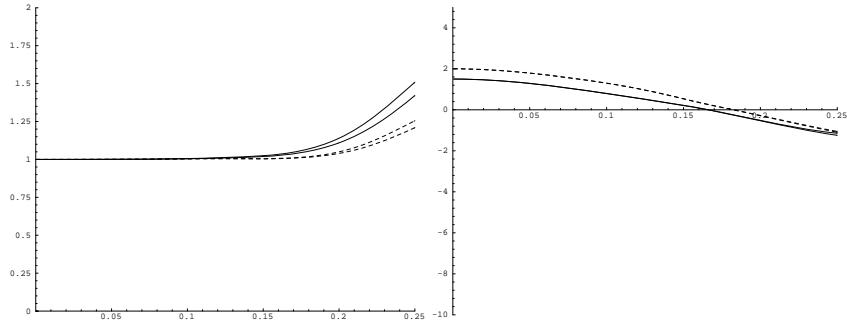


FIGURE 5. History of the x -coordinate (left) and the y -coordinate (right) of the centers of 2 circular particles obtained from different mesh sizes, $h_v = 1/192$ (thick lines) and $h_v = 1/256$ (thin lines).

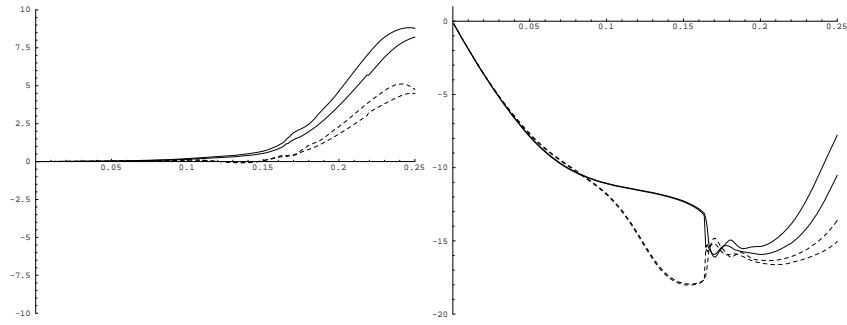


FIGURE 6. History of the x -component (left) and the y -component (right) of the translation velocity of 2 circular particles obtained from different mesh sizes, $h_v = 1/192$ (thick lines) and $h_v = 1/256$ (thin lines).

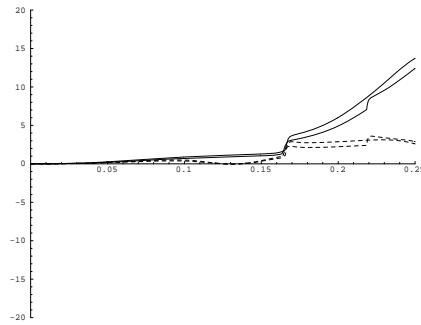


FIGURE 7. History of the angular speed of 2 circular particles obtained from different mesh sizes, $h_v = 1/192$ (thick lines) and $h_v = 1/256$ (thin lines).

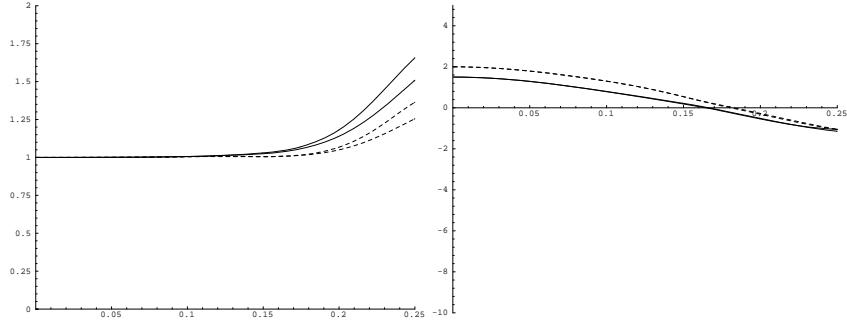


FIGURE 8. History of the x -coordinate (left) and the y -coordinate (right) of the centers of 2 circular particles obtained from different time steps, $\Delta t = 0.0005$ (thick lines) and $\Delta t = 0.00025$ (thin lines).

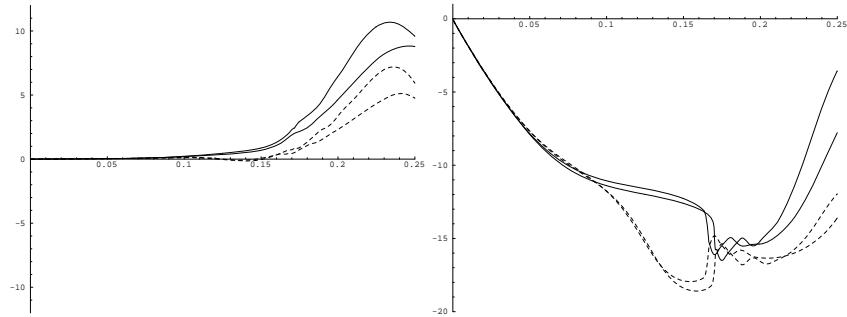


FIGURE 9. History of the x -component (left) and the y -component (right) of the (translation) velocity of 2 circular particles obtained from different time steps, $\Delta t = 0.0005$ (thick lines) and $\Delta t = 0.00025$ (thin lines).

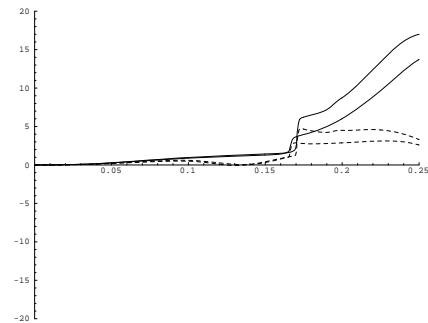


FIGURE 10. History of the angular speed of 2 circular particles obtained from different time steps, $\Delta t = 0.0005$ (thick lines) and $\Delta t = 0.00025$ (thin lines).

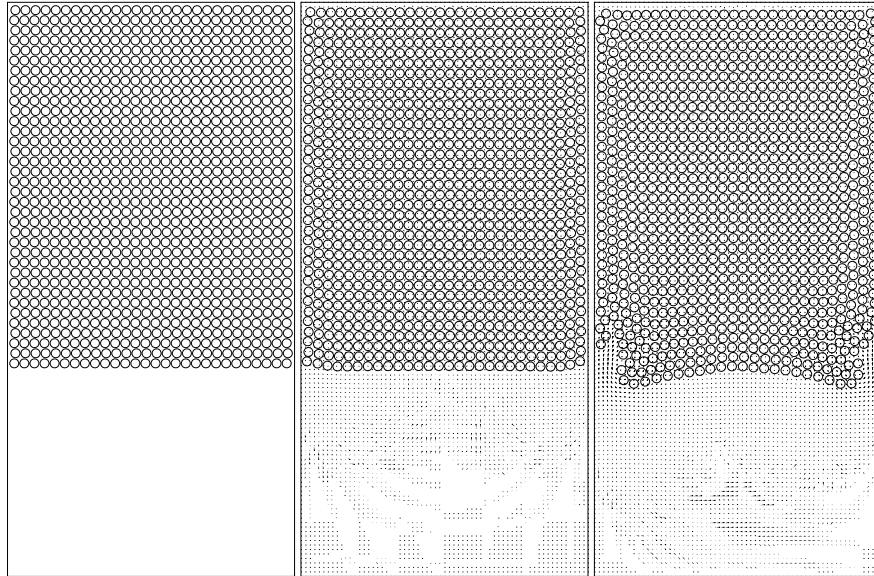


FIGURE 11. Sedimentation of 1008 circular particles: $t = 0, 1$, and 2 (left to right).

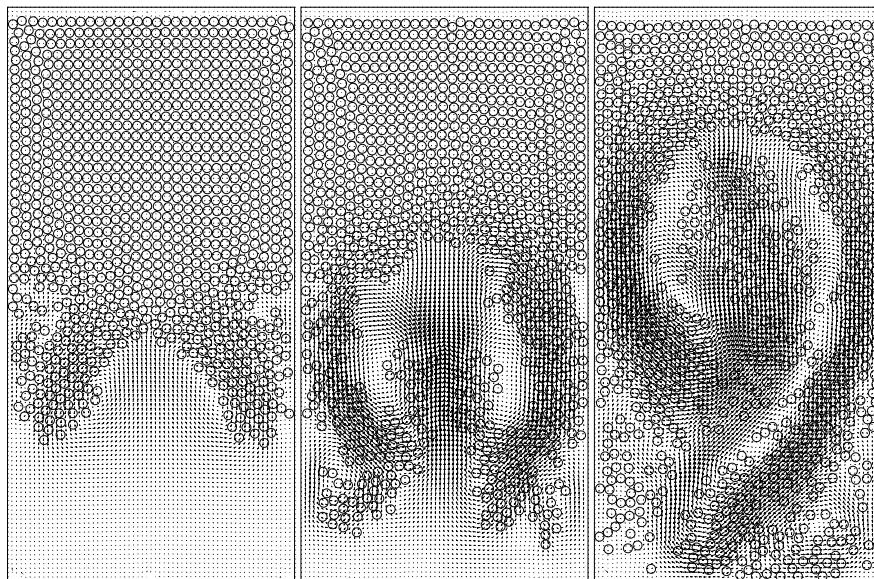


FIGURE 12. Sedimentation of 1008 circular particles: $t = 3, 4$, and 5 (left to right).

8.2. A 1008 particles case. The second test problem that we consider concerns the simulation of the motion of 1008 sedimenting cylinders in the closed channel, $\Omega = (0, 2) \times (0, 4)$. The diameter d of the cylinders is 0.0625 and the position of the cylinders at time $t = 0$ is shown in Figure 11. The solid fraction in this test

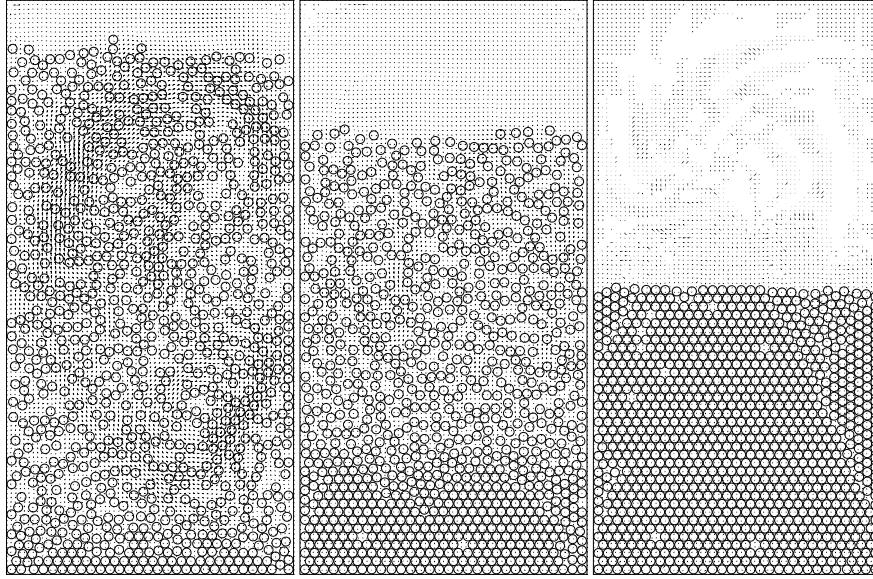


FIGURE 13. Sedimentation of 1008 circular particles: $t = 10, 20$, and 48 (left to right).

case is 38.66%. Initial velocity and angular speed of cylinders are $V_{p,i}^0 = \mathbf{0}$, $\omega_{p,i}^0 = 0$ for $i = 1, \dots, 1008$. The density of the fluid is $\rho_f = 1.0$ and the density of cylinders is $\rho_s = 1.01$. The viscosity of the fluid is $\nu_f = 0.01$. The initial condition for the fluid flow is $\mathbf{u} = \mathbf{0}$ and $\mathbf{g}_0(t) = \mathbf{0}, \forall t \geq 0$. The time step is $\Delta t = 0.001$. The mesh size for the velocity field is $h_v = 1/256$ (there are 525835 nodes). The mesh size for pressure is $h_p = 1/128$ (131841 nodes). For this many particles case, a fine mesh is required. The parameters for the repulsion force are $\rho = h_v$, $\epsilon_p = 3.26 \times 10^{-5}$, and $\epsilon_w = \epsilon_p/2$. We have chosen $\alpha = 1$ and $\beta = 0$ in the Marchuk-Yanenko scheme. The number of iterations for the divergence free projection problem varies from 12 to 14, the number of iterations for the linearized advection-diffusion problem is 5, and the one for the rigid body motion projection is about 7. Those number of iterations are almost independent of the mesh size and of the number of particles. With the finite dimensional spaces defined in Section 5, the evolution of the 1008 cylinders sedimenting in the closed channel is shown in Figures 11–13. The maximal particle Reynolds number in the entire evolution is 17.44. The slightly wavy shape of the interface observed at $t=1$ in Figure 11 is a typical onset of a Rayleigh-Taylor instability. When t is between 1 and 2, two small eddies are forming close to the left wall and the right wall and some particles are pulling down fast by these two eddies. Then other two stronger eddies are forming at the lower center of the channel for t between 2 and 4; they push some particles almost to the top wall of the channel. At the end all particles are settled at the bottom of the channel.

8.3. A three dimensional case. The third test problem that we consider here concerns the simulation of the motion of two sedimenting balls in a rectangular cylinder. The initial computational domain is $\Omega = (0, 1) \times (0, 1) \times (-1, 1.5)$, then it moves with the center of the lower ball. The diameter d of two balls is 0.25 and the position of balls at time $t = 0$ is shown in Figure 14. Initial velocity and angular

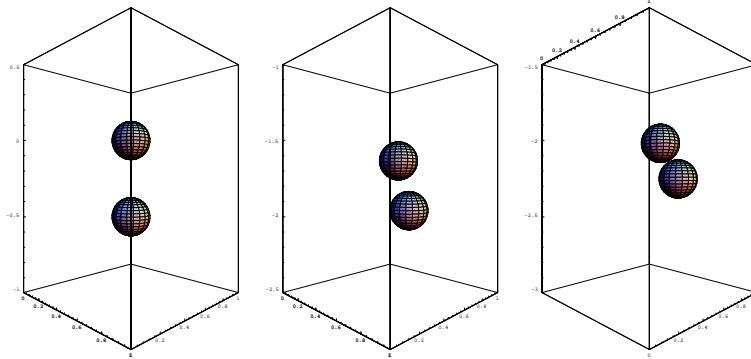


FIGURE 14. Sedimentation of two spherical particles: $t = 0.00$, 0.35 , and 0.40 (left to right)

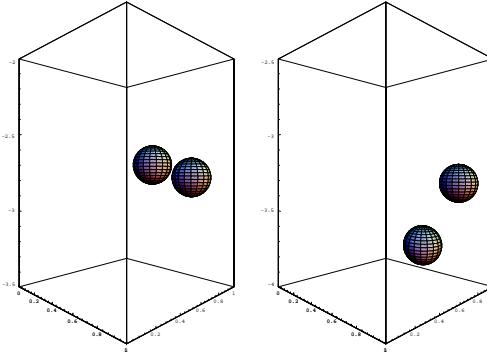


FIGURE 15. Sedimentation of two spherical particles: $t = 0.50$, and 0.70 (left to right)

speed of balls are zero. The density of the fluid is $\rho_f = 1.0$ and the density of balls is $\rho_s = 1.14$. The viscosity of the fluid is $\nu_f = 0.01$. The initial condition for the fluid flow is $\mathbf{u} = \mathbf{0}$. The mesh size for the velocity field is $h_v = 1/40$. The mesh size for pressure is $h_p = 1/20$. The time step is $\Delta t = 0.001$. For the repulsion force parameters, we have now taken, $\rho = h_v$, $\epsilon_p = 8.73 \times 10^{-3}$ and $\epsilon_w = \epsilon_p/2$. The maximal particle Reynolds number in the entire evolution is 198.8. In Figures 14 and 15, we can see the fundamental features of fluidizing two balls, i.e., drafting, kissing and tumbling.

9. Acknowledgments

We acknowledge the helpful comments and suggestions of E. J. Dean, V. Girault, J. He, Y. Kuznetsov, B. Maury, and G. Rodin and also the support of NEC concerning the use of an SX-3 supercomputer. We acknowledge the support of the NSF under HPCC Grand Challenge Grant ECS-9527123, NSF (Grants DMS 8822522, DMS 9112847, DMS 9217374), DRET (Grant 89424), DARPA (Contracts

AFOSR F49620-89-C-0125, AFOSR-90-0334), the Texas Board of Higher Education (Grants 003652156ARP and 003652146ATP) and the University of Houston (PEER grant 1-27682).

References

1. F. Bertrand, P.A. Tanguy, and F. Thibault, *A three-dimensional fictitious domain method for incompressible fluid flow problems*, Int. J. Num. Meth. Fluids **25** (1997), 615–631.
2. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, N.Y., 1991.
3. R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, N.Y., 1984.
4. R. Glowinski, T.I. Hesla, D.D. Joseph, T.W. Pan, and J. Périaux, *Distributed Lagrange multiplier methods for particulate flows*, Computational Science for the 21st Century (M.O. Bristeau, G. Etgen, W. Fitzgibbon, J.L. Lions, J. Périaux, and M.F. Wheeler, eds.), J. Wiley, Chichester, 1997, pp. 270–279.
5. R. Glowinski and P. LeTallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM, Philadelphia, PA, 1989.
6. R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux, *A distributed Lagrange multiplier/fictitious domain method for particulate flows*, submitted to *International J. of Multiphase Flow*.
7. T. I. Hesla, *The dynamical simulation of two-dimensional fluid/particle systems*, unpublished notes, 1991.
8. H. H. Hu, D. D. Joseph, and M. J. Crochet, *Direct simulation of fluid particle motions*, Theor. Comp. Fluid Dyn. **3** (1992), 285–306.
9. H.H. Hu, *Direct simulation of flows of solid-liquid mixtures*, Internat. J. Multiphase Flow **22** (1996), 335–352.
10. N. Kikuchi and T.J. Oden, *Contact problems in elasticity*, SIAM, Philadelphia, PA, 1988.
11. G. I. Marchuk, *Splitting and alternate direction methods*, Handbook of Numerical Analysis (P.G. Ciarlet and J.L. Lions, eds.), vol. 1, North-Holland, Amsterdam, 1990, pp. 197–462.
12. B. Maury, *A many-body lubrication model*, C.R. Acad. Sci., Paris, Série I (to appear).
13. B. Maury and R. Glowinski, *Fluid particle flow: a symmetric formulation*, C.R. Acad. Sci., Paris, Série I t. **324** (1997), 1079–1084.
14. O. Pironneau, J. Liou, and T. Tezduyar, *Characteristic-Galerkin and Galerkin least squares space-time formulations for advection-diffusion equations with time-dependent domains*, Comp. Meth. Appl. Mech. Eng. **16** (1992), 117–141.
15. J.E. Roberts and J.M. Thomas, *Mixed and hybrid methods*, Handbook of Numerical Analysis (P.G. Ciarlet and J.L. Lions, eds.), vol. 2, North-Holland, Amsterdam, 1991, pp. 521–639.
16. S. Turek, *A comparative study of time-stepping techniques for the incompressible navier-stokes equations: from fully implicit non-linear schemes to semi-implicit projection methods*, Int. J. Num. Meth. Fluids **22** (1996), 987–1011.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HOUSTON, HOUSTON, TEXAS 77204, USA
E-mail address: roland@math.uh.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HOUSTON, HOUSTON, TEXAS 77204, USA
E-mail address: pan@math.uh.edu

DEPARTMENT OF AEROSPACE ENGINEERING & MECHANICS, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MN 55455, USA
E-mail address: hesla@aem.umn.edu

DEPARTMENT OF AEROSPACE ENGINEERING & MECHANICS, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MN 55455, USA
E-mail address: joseph@aem.umn.edu

DASSAULT AVIATION, 92314 SAINT-CLOUD, FRANCE
E-mail address: periaux@menusin.inria.fr

Domain Decomposition Algorithms for Saddle Point Problems

Luca F. Pavarino

1. Introduction

In this paper, we introduce some domain decomposition methods for saddle point problems with or without a penalty term, such as the Stokes system and the mixed formulation of linear elasticity. We also consider more general nonsymmetric problems, such as the Oseen system, which are no longer saddle point problems but can be studied in the same abstract framework which we adopt.

Several approaches have been proposed in the past for the iterative solution of saddle point problems. We recall here:

- Uzawa's algorithm and its variants (Arrow, Hurwicz, and Uzawa [1], Elman and Golub [24], Bramble, Pasciak, and Vassilev [10], Maday, Meiron, Patera, and Rønquist [38]);
- multigrid methods (Verfürth [54], Wittum [55], Braess and Blömer [7], Brenner [11]);
- preconditioned conjugate gradient methods for a positive definite equivalent problem (Bramble and Pasciak [8]);
- block-diagonal preconditioners (Rusten and Winther [50], Silvester and Wathen [51], Klawonn [31]);
- block-triangular preconditioners (Elman and Silvester [25], Elman [23], Klawonn [32], Klawonn and Starke [34], Pavarino [43]).

Some of these approaches allow the use of domain decomposition techniques on particular subproblems, such as the inexact blocks in a block preconditioner. In this paper, we propose some alternative approaches based on the application of domain decomposition techniques to the whole saddle point problem, discretized with either h -version finite elements or spectral elements. We will consider both a) overlapping Schwarz methods and b) iterative substructuring methods. We refer to Smith, Bjørstad, and Gropp [52] or Chan and Mathew [18] for a general introduction to domain decomposition methods.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N35, 65N30, 65N22, 65F10.

This work was supported by I.A.N-CNR, Pavia and by the National Science Foundation under Grant NSF-CCR-9503408.

a) Early work by Lions [37] and Fortin and Aboulaich [29] extended the original overlapping Schwarz method to Stokes problems, but these methods were based on a positive definite problem obtained by working in the subspace of divergence-free functions and they did not have a coarse solver, which is essential for obtaining scalability. Later, the overlapping Schwarz method was also extended to the mixed formulations of scalar-second order elliptic problems (see Mathew [40, 41], Ewing and Wang [27], Rusten, Vassilevski, and Winther [49]) and to indefinite, nonsymmetric, scalar-second order elliptic problems (see Cai and Widlund [13, 14]). In Section 6, we present a different overlapping Schwarz method based on the solution of local saddle point problems on overlapping subdomains and the solution of a coarse saddle point problem. The iteration is accelerated by a Krylov space method, such as GMRES or QMR. The resulting method is the analog for saddle point problems of the method proposed and analyzed by Dryja and Widlund [20, 21] for symmetric positive definite elliptic problems. As in the positive definite case, our method is parallelizable, scalable, and has a simple coarse problem. This work on overlapping methods is joint with Axel Klawonn of the Westfälische Wilhelms-Universität Münster, Germany.

b) Nonoverlapping domain decomposition preconditioners for Stokes problems have been considered by Bramble and Pasciak [9], Quarteroni [47] and for spectral element discretizations by Fischer and Rønquist [28], Rønquist [48], Le Tallec and Patra [36], and Casarin [17]. In Section 7, we present a class of iterative substructuring methods in which the saddle point Schur complement, obtained after the elimination of the internal velocities and pressures in each subdomain, is solved with a block preconditioner. The velocity block can be constructed using wire basket or Neumann-Neumann techniques. In the Stokes case, this construction is directly based on the original scalar algorithms, while in the elasticity case it requires an extension of the scalar techniques. The iteration is accelerated by a Krylov space method, such as GMRES or PCR. The resulting algorithms are parallelizable and scalable, but the structure of the coarse problem is more complex than in overlapping methods. This work on nonoverlapping methods is joint with Olof B. Widlund of the Courant Institute, New York University, USA.

2. Model saddle point problems

The Stokes system. Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ be a polyhedral domain and $L_0^2(\Omega)$ be the subset of $L^2(\Omega)$ consisting of functions with zero mean value. Given $\mathbf{f} \in (H^{-1}(\Omega))^d$ and, for simplicity, homogeneous Dirichlet boundary conditions, the Stokes problem consists in finding the velocity $\mathbf{u} \in \mathbf{V} = (H_0^1(\Omega))^d$ and the pressure $p \in U = L_0^2(\Omega)$ of an incompressible fluid with viscosity μ by solving:

$$(1) \quad \begin{cases} \mu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} dx - \int_{\Omega} \operatorname{div} \mathbf{v} p dx &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} dx \quad \forall \mathbf{v} \in \mathbf{V}, \\ - \int_{\Omega} \operatorname{div} \mathbf{u} q dx &= 0 \quad \forall q \in U. \end{cases}$$

Linear elasticity in mixed form. The following mixed formulation of the system of linear elasticity describes the displacement \mathbf{u} and the variable $p = -\lambda \operatorname{div} \mathbf{u}$ of an almost incompressible material with Lamé constants λ and μ . The material is fixed along $\Gamma_0 \subset \partial\Omega$, subject to a surface force of density \mathbf{g} along $\Gamma_1 = \partial\Omega \setminus \Gamma_0$ and

subject to an external force \mathbf{f} :

$$(2) \quad \begin{cases} 2\mu \int_{\Omega} \epsilon(\mathbf{u}) : \epsilon(\mathbf{v}) \, dx - \int_{\Omega} \operatorname{div} \mathbf{v} p \, dx = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{V}, \\ - \int_{\Omega} \operatorname{div} \mathbf{u} q \, dx - \frac{1}{\lambda} \int_{\Omega} pq \, dx = 0 \quad \forall q \in L^2(\Omega). \end{cases}$$

Here $\mathbf{V} = \{\mathbf{v} \in H^1(\Omega)^d : \mathbf{v}|_{\Gamma_0} = 0\}$, $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ are the components of the linearized strain tensor $\epsilon(\mathbf{u})$, and $\langle \mathbf{F}, \mathbf{v} \rangle = \int_{\Omega} \sum_{i=1}^3 f_i v_i \, dx + \int_{\Gamma_1} \sum_{i=1}^3 g_i v_i \, ds$. It is well known that this mixed formulation is a good remedy for the locking and ill-conditioning problems that arise in the pure displacement formulation when the material becomes almost incompressible; see Babuška and Suri [2]. The incompressibility of the material can be characterized by λ approaching infinity or, equivalently, by the Poisson ratio $\nu = \frac{\lambda}{2(\lambda+\mu)}$ approaching 1/2.

The Oseen system (linearized Navier-Stokes). An example of nonsymmetric problem is given by the Oseen system. Linearizing the Navier-Stokes equations by a fixed-point or Picard iteration, we have to solve in each step the following Oseen problem: given a divergence-free vector field \mathbf{w} , find the velocity $\mathbf{u} \in \mathbf{V} = (H_0^1(\Omega))^d$ and the pressure $p \in U = L_0^2(\Omega)$ of an incompressible fluid with viscosity μ satisfying

$$(3) \quad \begin{cases} \mu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} dx + \int_{\Omega} [(\mathbf{w} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} dx - \int_{\Omega} \operatorname{div} \mathbf{v} p dx = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} dx \quad \forall \mathbf{v} \in \mathbf{V}, \\ - \int_{\Omega} \operatorname{div} \mathbf{u} q dx = 0 \quad \forall q \in U. \end{cases}$$

Here the right-hand side \mathbf{f} is as in the Stokes problem and the convection term is given by the skew-symmetric bilinear form $\int_{\Omega} [(\mathbf{w} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} dx = \int_{\Omega} \sum_{i,j=1}^3 w_j \frac{\partial u_i}{\partial x_j} v_i dx$.

3. An abstract framework for saddle point problems and generalizations

In general, given two Hilbert spaces \mathbf{V} and U , the algorithms described in this paper apply to the following generalization of abstract saddle point problems with a penalty term. An analysis and a more complete treatment can be found in Brezzi and Fortin [12].

Find $(\mathbf{u}, p) \in \mathbf{V} \times U$ such that

$$(4) \quad \begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) - t^2 c(p, q) = \langle G, q \rangle \quad \forall q \in U \quad t \in [0, 1], \end{cases}$$

where $\mathbf{F} \in \mathbf{V}'$ and $G \in U'$. When $a(\cdot, \cdot)$ and $c(\cdot, \cdot)$ are symmetric, (4) is a saddle point problem, but we keep the same terminology also in the more general nonsymmetric case. In order to have a well-posed problem, we assume that the following properties are satisfied. Let $B : \mathbf{V} \rightarrow U'$ and its transpose $B^T : U \rightarrow \mathbf{V}'$ be the linear operators defined by

$$(B\mathbf{v}, q)_{U' \times U} = (\mathbf{v}, B^T q)_{V \times V'} = b(\mathbf{v}, q) \quad \forall \mathbf{v} \in \mathbf{V}, \forall q \in U.$$

i) $a(\cdot, \cdot) : \mathbf{V} \times \mathbf{V} \longrightarrow R$ is a continuous, positive semidefinite bilinear form, invertible on the kernel $\text{Ker } B$ of B , i.e.

$$\exists \alpha_0 > 0 \text{ such that } \begin{cases} \inf_{\mathbf{u} \in \text{Ker } B} \sup_{\mathbf{v} \in \text{Ker } B} \frac{a(\mathbf{u}, \mathbf{v})}{\|\mathbf{u}\|_V \|\mathbf{v}\|_V} \geq \alpha_0, \\ \inf_{\mathbf{v} \in \text{Ker } B} \sup_{\mathbf{u} \in \text{Ker } B} \frac{a(\mathbf{u}, \mathbf{v})}{\|\mathbf{u}\|_V \|\mathbf{v}\|_V} \geq \alpha_0; \end{cases}$$

ii) $b(\cdot, \cdot) : \mathbf{V} \times U \longrightarrow R$ is a continuous bilinear form satisfying the inf-sup condition

$$\exists \beta_0 > 0 \text{ such that } \sup_{v \in V} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_V} \geq \beta_0 \|q\|_{U/\text{Ker } B^T};$$

iii) $c(\cdot, \cdot) : U \times U \longrightarrow R$ is a symmetric, continuous, U -elliptic bilinear form.

More general conditions could be assumed; see Brezzi and Fortin [12] and Braess [6].

For simplicity, we adopt in the following the Stokes terminology, i.e. we call the variables in \mathbf{V} velocities and the variables in U pressures.

4. Mixed finite element methods: $P_1(h) - P_1(2h)$ and $Q_1(h) - P_0(h)$ stabilized

The continuous problem (4) is discretized by introducing finite element spaces $\mathbf{V}^h \subset \mathbf{V}$ and $U^h \subset U$. For simplicity, we consider uniform meshes, but more general nonuniform meshes may be used. We consider two choices of finite element spaces, in order to illustrate our algorithms for both stable and stabilized discretizations, with continuous and discontinuous pressures respectively.

a) $P_1(h) - P_1(2h)$ (also known as $P2 - iso - P1$). Let τ_{2h} be a triangular finite element mesh of Ω of characteristic mesh size $2h$ and let τ_h be a refinement of τ_{2h} . We introduce finite element spaces consisting of continuous piecewise linear velocities on τ_h and continuous piecewise linear pressures on τ_{2h} :

$$\begin{aligned} \mathbf{V}^h &= \{\mathbf{v} \in (C(\Omega))^d \cap \mathbf{V} : \mathbf{v}|_T \in P_1, T \in \tau_h\}, \\ U^h &= \{q \in C(\Omega) \cap U : q|_T \in P_1, T \in \tau_{2h}\}. \end{aligned}$$

This is a stable mixed finite element method, i.e. it satisfies a uniform inf-sup condition (see Brezzi and Fortin [12]).

b) $Q_1(h) - P_0(h)$ stabilized. Here the velocities are continuous piecewise trilinear (bilinear in 2D) functions on a quadrilateral mesh of size h and the pressures are piecewise constant (discontinuous) functions on the same mesh:

$$\begin{aligned} \mathbf{V}^h &= \{\mathbf{v} \in (C(\Omega))^d \cap \mathbf{V} : \mathbf{v}|_T \in Q_1, T \in \tau_h\}, \\ U^h &= \{q \in U : q|_T \in P_0, T \in \tau_h\}. \end{aligned}$$

This couple of finite element spaces does not satisfy the inf-sup condition, but can be stabilized as shown in Kechkar and Silvester [30] by relaxing the discrete incompressibility condition. In the Stokes case in two dimensions, this stabilization is achieved by defining a nonoverlapping macroelement partitioning \mathcal{M}_h such that each macroelement $M \in \mathcal{M}_h$ is a connected set of adjoining elements from τ_h . Denoting by Γ_M the set of interelement edges in the interior of M and by $e \in \Gamma_M$ one of these interior edges, the original bilinear form $c(p, q) = 0$ is replaced by

$$(5) \quad c_h(p, q) = \beta \sum_{M \in \mathcal{M}_h} \sum_{e \in \Gamma_M} h_e \int_e [\![p]\!]_e [\![q]\!]_e ds.$$

Here $\llbracket p \rrbracket_e$ is the jump operator across $e \in \Gamma_M$, h_e is the length of e , and β is a stabilization parameter; see [30] for more details and an analysis.

By discretizing the saddle point problem (4) with these mixed finite elements, we obtain the following discrete saddle point problem:

$$(6) \quad K_h x = \begin{bmatrix} A_h & B_h^T \\ B_h & -t^2 C_h \end{bmatrix} x = f_h .$$

The matrix K_h is symmetric and indefinite whenever A_h is symmetric, as in the Stokes and elasticity cases. The penalty parameter t^2 is zero in the Stokes, Oseen and incompressible elasticity cases when discretized with stable elements, such as $P_1(h) - P_1(2h)$; it is nonzero in the case of almost incompressible elasticity or when stabilized elements, such as $Q_1(h) - P_0(h)$ stabilized, are used.

5. Mixed spectral element methods: $Q_n - Q_{n-2}$ and $Q_n - P_{n-1}$

The continuous problem (4) can also be discretized by conforming spectral elements. Let Ω_{ref} be the reference cube $(-1, 1)^3$, let $Q_n(\Omega_{\text{ref}})$ be the set of polynomials on Ω_{ref} of degree n in each variable, and let $P_n(\Omega_{\text{ref}})$ be the set of polynomials on Ω_{ref} of total degree n . We assume that the domain Ω can be decomposed into N nonoverlapping finite elements Ω_i , each of which is an affine image of the reference cube. Thus, $\Omega_i = \phi_i(\Omega_{\text{ref}})$, where ϕ_i is an affine mapping.

a) $Q_n - Q_{n-2}$. This method was proposed by Maday, Patera, and Rønquist [39] for the Stokes system. \mathbf{V} is discretized, component by component, by continuous, piecewise polynomials of degree n :

$$\mathbf{V}^n = \{\mathbf{v} \in \mathbf{V} : v_k|_{\Omega_i} \circ \phi_i \in Q_n(\Omega_{\text{ref}}), i = 1, \dots, N, k = 1, 2, 3\}.$$

The pressure space is discretized by piecewise polynomials of degree $n - 2$:

$$U^n = \{q \in U : q|_{\Omega_i} \circ \phi_i \in Q_{n-2}(\Omega_{\text{ref}}), i = 1, \dots, N\}.$$

We note that the elements of U^n are discontinuous across the boundaries of the elements Ω_i . These mixed spectral elements are implemented using Gauss-Lobatto-Legendre (GLL) quadrature, which also allows the construction of a very convenient tensor-product basis for \mathbf{V}^n . Denote by $\{\xi_i, \xi_j, \xi_k\}_{i,j,k=0}^n$ the set of GLL points of $[-1, 1]^3$, and by σ_i the quadrature weight associated with ξ_i . Let $l_i(x)$ be the Lagrange interpolating polynomial of degree n which vanishes at all the GLL nodes except ξ_i , where it equals one. Each element of $Q_n(\Omega_{\text{ref}})$ is expanded in the GLL basis

$$u(x, y, z) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n u(\xi_i, \xi_j, \xi_k) l_i(x) l_j(y) l_k(z),$$

and each L^2 -inner product of two scalar components u and v is replaced by

$$(7) \quad (u, v)_{n,\Omega} = \sum_{s=1}^N \sum_{i,j,k=0}^n (u \circ \phi_s)(\xi_i, \xi_j, \xi_k) (v \circ \phi_s)(\xi_i, \xi_j, \xi_k) |J_s| \sigma_i \sigma_j \sigma_k,$$

where $|J_s|$ is the determinant of the Jacobian of ϕ_s . Similarly, a very convenient basis for U^n consists of the tensor-product Lagrangian nodal basis functions associated with the internal GLL nodes. Another basis associated with the Gauss-Legendre (GL) nodes has been studied in [28] and [38]. We refer to Bernardi and Maday [3, 4] for more details and the analysis of the resulting discrete problem.

The $Q_n - Q_{n-2}$ method satisfies the nonuniform inf-sup condition

$$(8) \quad \sup_{\mathbf{v} \in \mathbf{V}^n} \frac{(\operatorname{div} \mathbf{v}, q)}{\|\mathbf{v}\|_{H^1}} \geq C n^{-(\frac{d-1}{2})} \|q\|_{L^2} \quad \forall q \in U^n,$$

where $d = 2, 3$ and the constant C is independent of n and q ; see Maday, Patera, and Rønquist [39] and Stenberg and Suri [53]. However, numerical experiments, reported in Maday, Meiron, Patera, and Rønquist, see [38] and [39], have also shown that for practical values of n , e.g., $n \leq 16$, the inf-sup constant β_n of the $Q_n - Q_{n-2}$ method decays much slower than what would be expected from the theoretical bound.

b) $Q_n - P_{n-1}$. This method uses the same velocity space \mathbf{V}^n described before, together with an alternative pressure space given by piecewise polynomials of total degree $n - 1$:

$$\{q \in U : q|_{\Omega_i} \circ \phi_i \in P_{n-1}(\Omega_{ref}), i = 1, \dots, N\}.$$

This choice has been studied by Stenberg and Suri [53] and more recently by Bernardi and Maday [5], who proved a uniform inf-sup condition for it. Its practical application is limited by the lack of a standard tensorial basis for P_{n-1} ; however, other bases, common in the p -version finite element literature, can be used.

Other interesting choices for U^n have been studied in Canuto [15] and Canuto and Van Kemenade [16] in connection with stabilization techniques for spectral elements using bubble functions.

Applying GLL quadrature to the abstract problem (4), we obtain again a discrete saddle point problem of the form

$$(9) \quad K_n x = \begin{bmatrix} A_n & B_n^T \\ B_n & -t^2 C_n \end{bmatrix} x = f_n .$$

As before, K_n is a symmetric indefinite matrix in the Stokes and elasticity case, while it is a nonsymmetric matrix in the Oseen case.

6. Overlapping Schwarz Methods

We present here the basic idea of the method for the additive variant of the preconditioner and $P_1(h) - P_1(2h)$ finite elements on uniform meshes (see Section 4). More general multiplicative or hybrid variants, unstructured meshes and spectral element discretizations can be considered as well. See Klawonn and Pavarino [33] for a more complete treatment.

Let τ_H be a coarse finite element triangulation of the domain Ω into N subdomains Ω_i of characteristic diameter H . A fine triangulation τ_h is obtained as a refinement of τ_H and H/h will denote the number of nodes on each subdomain side. In order to have an overlapping partition of Ω , each subdomain Ω_i is extended to a larger subdomain Ω'_i , consisting of all elements of τ_h within a distance δ from Ω_i .

Our overlapping additive Schwarz preconditioner \widehat{K}_{OAS}^{-1} for K_h is based on the solutions of local saddle point problems on the subdomains Ω'_i and on the solution of a coarse saddle point problem on the coarse mesh τ_H . In matrix form:

$$(10) \quad \widehat{K}_{OAS}^{-1} = R_0^T K_0^{-1} R_0 + \sum_{i=1}^N R_i^T K_i^{-1} R_i,$$

where $R_0^T K_0^{-1} R_0$ represents the coarse problem and $R_i^T K_i^{-1} R_i$ represents the i -th local problem. This preconditioner is associated with the following decomposition

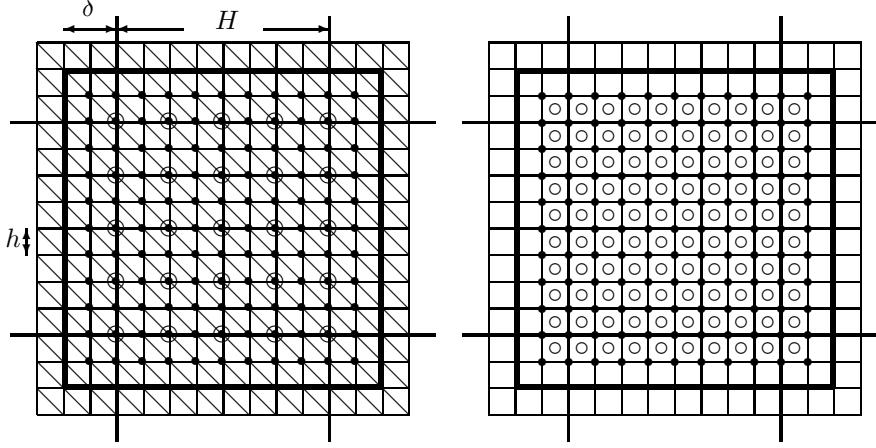


FIGURE 1. Local spaces associated with an interior subdomain Ω'_i . $P_1(h) - P_1(2h)$ (left) and $Q_1(h) - P_0(h)$ stabilized (right): velocity degrees of freedom are denoted by bullets (\bullet), pressure degrees of freedom are denoted by circles (\circ). Subdomain size $H/h = 8$, overlap $\delta = 2h$.

of the discrete space $\mathbf{V}^h \times U^h$ into a coarse space $\mathbf{V}_0^h \times U_0^h$ and local spaces $\mathbf{V}_i^h \times U_i^h$, associated with the subdomains Ω'_i :

$$\mathbf{V}^h \times U^h = \mathbf{V}_0^h \times U_0^h + \sum_{i=1}^N \mathbf{V}_i^h \times U_i^h.$$

a) *Coarse problem.* For $P_1(h) - P_1(2h)$ elements, the coarse space is defined as

$$\mathbf{V}_0^h = \mathbf{V}^{H/2}, \quad U_0^h = U^{H/2}.$$

The associated coarse stiffness matrix is $K_0 = K_{H/2}$, obtained using $P_1(H/2) - P_1(H)$ mixed elements and R_0^T represents the standard piecewise bilinear interpolation matrix between coarse and fine degrees of freedom, for both velocities and pressures. We use $H/2$ as the mesh size of the coarse velocities because we choose H as the mesh size of the coarse pressures.

For $Q_1(h) - P_0(h)$ stabilized elements, the coarse space is defined as

$$V_0^h = V^H, \quad U_0^h = U^H.$$

The associated coarse stiffness matrix is $K_0 = K_H$ and R_0^T is the standard piecewise bilinear interpolation matrix between coarse and fine velocities and the standard injection matrix between coarse and fine pressures.

b) *Local problems.* For $P_1(h) - P_1(2h)$ finite elements (with continuous pressures), the local spaces consist of velocities and zero mean value pressures satisfying zero Dirichlet boundary conditions on the internal subdomain boundaries $\partial\Omega'_i \setminus \partial\Omega$:

$$\mathbf{V}_i^h = \mathbf{V}^h \cap (H_0^1(\Omega'_i))^d,$$

$$U_i^h = \{q \in U^h \cap L_0^2(\Omega'_i) : q = 0 \text{ on } \partial\Omega'_i \setminus \partial\Omega \text{ and outside } \Omega'_i\}.$$

Here, the minimal overlap is one pressure element, i.e. $\delta = 2h$.

For $Q_1(h) - P_0(h)$ stabilized elements, the pressures are discontinuous piecewise constant functions and there are no degrees of freedom associated with $\partial\Omega'_i \setminus \partial\Omega$. In this case, we set to zero the pressure degrees of freedom in the elements that touch $\partial\Omega'_i \setminus \partial\Omega$. The associated local pressure spaces are:

$$U_i^h = \{q \in L_0^2(\Omega'_i) : q|_T = 0 \ \forall T : \bar{T} \cap (\partial\Omega'_i \setminus \partial\Omega) \neq \emptyset\}.$$

Here, the minimal overlap is $\delta = h$. In matrix terms, the matrices R_i in (10) are restriction matrices returning the degrees of freedom associated with the interior of Ω'_i and $K_i = R_i K_h R_i^T$ are the local stiffness matrices. Each discrete local problem (and its matrix representation K_i) is nonsingular because of the zero mean-value constraint for the local pressure solution. See Figure 1 for a graphic representation of these local spaces in two dimensions.

We remark that \hat{K}_{OAS}^{-1} is a nonsingular preconditioner, since K_0 and $K_i, i = 1, \dots, N$, are nonsingular matrices. In the symmetric cases (Stokes and elasticity), \hat{K}_{OAS}^{-1} is a symmetric indefinite preconditioner. If we need to work with global zero mean-value pressures, as in the Stokes and Oseen problems or in the incompressible limit of the mixed linear elasticity problem, we enforce this constraint in each application of the preconditioner.

7. Iterative substructuring methods for spectral element discretizations

The elimination of the interior unknowns in a saddle point problem is somewhat different than the analogous process in a positive definite problem. In this section, we illustrate this process for the spectral element discretization (see Section 5) of the Stokes problem. We will see that the remaining interface unknowns and constant pressures in each spectral element satisfy a reduced saddle point problem, analogous to the Schur complement in the positive definite case. We refer to Pavarino and Widlund [45] for a more complete treatment.

The interface Γ of the decomposition $\{\Omega_i\}$ of Ω is defined by

$$\Gamma = (\cup_{i=1}^N \partial\Omega_i) \setminus \partial\Omega.$$

The discrete space of restrictions to the interface is defined by

$$\mathbf{V}_\Gamma^n = \{\mathbf{v}|_\Gamma, \mathbf{v} \in \mathbf{V}^n\}.$$

Γ is composed of N_F faces F_k (open sets) of the elements and the wire basket W , defined as the union of the edges and vertices of the elements, i.e.

$$(11) \quad \Gamma = \cup_{k=1}^{N_F} F_k \cup W.$$

We first define local subspaces consisting of velocities with support in the interior of individual elements,

$$(12) \quad \mathbf{V}_i^n = \mathbf{V}^n \cap H_0^1(\Omega_i)^3, \quad i = 1, \dots, N,$$

and local subspaces consisting of pressures with support and zero mean value in individual elements

$$(13) \quad U_i^n = U^n \cap L_0^2(\Omega_i), \quad i = 1, \dots, N.$$

The velocity space \mathbf{V}^n is decomposed as

$$\mathbf{V}^n = \mathbf{V}_1^n + \mathbf{V}_2^n + \dots + \mathbf{V}_N^n + \mathbf{V}_S^n,$$

where the local spaces \mathbf{V}_i^n have been defined in (12) and

$$(14) \quad \mathbf{V}_S^n = \mathcal{S}^n(\mathbf{V}_\Gamma^n)$$

is the subspace of interface velocities. The discrete Stokes extension \mathcal{S}^n is the operator that maps any $\mathbf{u} \in \mathbf{V}_\Gamma^n$ into the velocity component of the solution of the following Stokes problem on each element:

Find $\mathcal{S}^n \mathbf{u} \in \mathbf{V}^n$ and $p \in (\sum_{i=1}^N U_i^n)$ such that on each Ω_i

$$(15) \quad \begin{cases} s_n(\mathcal{S}^n \mathbf{u}, \mathbf{v}) + b_n(\mathbf{v}, p) = 0 & \forall \mathbf{v} \in \mathbf{V}_i^n \\ b_n(\mathcal{S}^n \mathbf{u}, q) = 0 & \forall q \in U_i^n \\ \mathcal{S}^n \mathbf{u} = \mathbf{u} \text{ on } \partial \Omega_i. \end{cases}$$

Here the discrete bilinear forms are $s_n(\mathbf{u}, \mathbf{v}) = \mu(\nabla \mathbf{u} : \nabla \mathbf{v})_{n,\Omega}$ and $b_n(\mathbf{u}, p) = -(\operatorname{div} \mathbf{u}, p)_{n,\Omega}$, where the discrete L^2 -inner product has been defined in (7). In the elasticity case, an analogous interface space \mathbf{V}_M^n can be defined using a discrete mixed elasticity extension operator. The pressure space U^n is decomposed as

$$U^n = U_1^n + U_2^n + \cdots + U_N^n + U_0,$$

where the local spaces U_i^n have been defined in (13) and

$$U_0 = \{q \in U^n : q|_{\Omega_i} = \text{constant}, i = 1, \dots, N\}$$

consists of piecewise constant pressures in each element. The vector of unknowns is now reordered placing first the interior unknowns, element by element, and then the interface velocities and the piecewise constant pressures in each element:

$$(\mathbf{u}, p)^T = (\mathbf{u}_1 \ p_1, \mathbf{u}_2 \ p_2, \dots, \mathbf{u}_N \ p_N, \mathbf{u}_\Gamma \ p_0)^T.$$

After this reordering, our saddle point problem (9) has the following matrix structure:

$$(16) \quad \left[\begin{array}{cccccc} A_{11} & B_{11}^T & \cdots & 0 & 0 & A_{1\Gamma} & 0 \\ B_{11} & 0 & \cdots & 0 & 0 & B_{1\Gamma} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{NN} & B_{NN}^T & A_{N\Gamma} & 0 \\ 0 & 0 & \cdots & B_{NN} & 0 & B_{N\Gamma} & 0 \\ A_{\Gamma 1} & B_{1\Gamma}^T & \cdots & A_{\Gamma N} & B_{N\Gamma}^T & A_{\Gamma\Gamma} & B_0^T \\ 0 & 0 & \cdots & 0 & 0 & B_0 & 0 \end{array} \right] \begin{bmatrix} \mathbf{u}_1 \\ p_1 \\ \vdots \\ \mathbf{u}_N \\ p_N \\ \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ 0 \\ \vdots \\ \mathbf{b}_N \\ 0 \\ \mathbf{b}_\Gamma \\ 0 \end{bmatrix}.$$

The leading block of this matrix is the direct sum of N local saddle point problems for the interior velocities and pressures (\mathbf{u}_i, p_i) . In addition there is a reduced saddle point problem for the interface velocities and piecewise constant pressures (\mathbf{u}_Γ, p_0) . These subsystems are given by

$$(17) \quad \begin{cases} A_{ii}\mathbf{u}_i + B_{ii}^T p_i = \mathbf{b}_i - A_{i\Gamma}\mathbf{u}_\Gamma & i = 1, 2, \dots, N, \\ B_{ii}\mathbf{u}_i = -B_{i\Gamma}\mathbf{u}_\Gamma \end{cases}$$

and

$$(18) \quad \begin{cases} A_{\Gamma\Gamma}\mathbf{u}_\Gamma + A_{\Gamma 1}\mathbf{u}_1 + \cdots + A_{\Gamma N}\mathbf{u}_N + B_{1\Gamma}^T p_1 + \cdots + B_{N\Gamma}^T p_N + B_0^T p_0 = \mathbf{b}_\Gamma \\ B_0\mathbf{u}_\Gamma = 0. \end{cases}$$

The local saddle point problems (17) are uniquely solvable because the local pressures are constrained to have zero mean value. The reduced saddle point problem

(18) can be written more clearly by introducing the linear operators R_i^b, R_i^Γ and P_i^b, P_i^Γ representing the solutions of the i -th local saddle point problem:

$$\mathbf{u}_i = R_i^b \mathbf{b}_i + R_i^\Gamma \mathbf{u}_\Gamma, \quad p_i = P_i^b \mathbf{b}_i + P_i^\Gamma \mathbf{u}_\Gamma, \quad i = 1, 2, \dots, N.$$

Then (18) can be rewritten as

$$(19) \quad \begin{cases} S_\Gamma \mathbf{u}_\Gamma + B_0^T p_0 = \tilde{\mathbf{b}}_\Gamma \\ B_0 \mathbf{u}_\Gamma = 0, \end{cases}$$

where

$$S_\Gamma = A_{\Gamma\Gamma} + \sum_{i=1}^N A_{\Gamma i} R_i^\Gamma + \sum_{i=1}^N B_{i\Gamma}^T P_i^\Gamma, \quad \tilde{\mathbf{b}}_\Gamma = \mathbf{b}_\Gamma - \sum_{i=1}^N A_{\Gamma i} R_i^b \mathbf{b}_i - \sum_{i=1}^N B_{i\Gamma}^T P_i^b \mathbf{b}_i.$$

As always, the matrices R_i^b, R_i^Γ and P_i^b, P_i^Γ need not be assembled explicitly; their action on given vectors is computed by solving the corresponding local saddle point problem. Analogously, S_Γ need not be assembled, since its action on a given vector can be computed by solving the N local saddle point problems (17) with $\mathbf{b}_i = 0$. The right-hand side $\tilde{\mathbf{b}}_\Gamma$ is formed from an additional set of solutions of the N local saddle point problems (17) with $\mathbf{u}_\Gamma = 0$.

The saddle point Schur complement (19) satisfies a uniform inf-sup condition (see [45] for a proof):

LEMMA 1.

$$\sup_{\mathcal{S}^n \mathbf{v} \in \mathbf{V}_S^n} \frac{(\operatorname{div} \mathcal{S}^n \mathbf{v}, q_0)^2}{s_n(\mathcal{S}^n \mathbf{v}, \mathcal{S}^n \mathbf{v})} \geq \beta_\Gamma^2 \|q_0\|_{L^2}^2 \quad \forall q_0 \in U_0,$$

where β_Γ is independent of q_0, n , and N .

An analogous stability result holds for the incompressible elasticity case.

7.1. Block preconditioners for the saddle point Schur complement.

We solve the saddle point Schur complement system (19) by some preconditioned Krylov space method such as PCR, if we use a symmetric positive definite preconditioner and the problem is symmetric, or GMRES if we use a more general preconditioner. Let S be the coefficient matrix of the reduced saddle point problem (19)

$$(20) \quad S = \begin{bmatrix} S_\Gamma & B_0^T \\ B_0 & 0 \end{bmatrix}.$$

We will consider the following block-diagonal and lower block-triangular preconditioners (an upper block-triangular preconditioner could be considered as well):

$$\widehat{S}_D = \begin{bmatrix} \widehat{S}_\Gamma & 0 \\ 0 & \widehat{C}_0 \end{bmatrix} \quad \widehat{S}_T = \begin{bmatrix} \widehat{S}_\Gamma & 0 \\ B_0 & -\widehat{C}_0 \end{bmatrix},$$

where \widehat{S}_Γ and \widehat{C}_0 are good preconditioners for S_Γ and the coarse pressure mass matrix C_0 , respectively. We refer to Klawonn [31, 32] for an analysis of block preconditioners. We consider two choices for \widehat{S}_Γ , based on wire basket and Neumann-Neumann techniques, and we take $\widehat{C}_0 = C_0$.

a) *A wire basket preconditioner for Stokes problems.* We first consider a simple Laplacian-based wire basket preconditioner

$$(21) \quad \widehat{S}_\Gamma = \begin{bmatrix} \widehat{S}_W & 0 & 0 \\ 0 & \widehat{S}_W & 0 \\ 0 & 0 & \widehat{S}_W \end{bmatrix},$$

where we use on each scalar component the scalar wire basket preconditioner introduced in Pavarino and Widlund [44] and extended to GLL quadrature based approximations in [46],

$$\widehat{S}_W^{-1} = R_0 \widehat{S}_{WW}^{-1} R_0^T + \sum_{k=1}^{N_F} R_{F_k} S_{F_k F_k}^{-1} R_{F_k}^T.$$

Here R_0 is a matrix representing a change of basis in the wire basket space, $R_{F_k}^T$ are restriction matrices returning the degrees of freedom associated with the face F_k , $k = 1, \dots, N_F$, and \widehat{S}_{WW} is an approximation of the original wire basket block. This is an additive preconditioner with independent parts associated with each face and the wire basket of the elements, defined in (11). It satisfies the following bound, proven in [45].

THEOREM 2. *Let the blocks of the block-diagonal preconditioner \widehat{S}_D be the wire basket preconditioner \widehat{S}_Γ defined in (21) and the coarse mass matrix C_0 . Then the Stokes saddle point Schur complement S preconditioned by \widehat{S}_D satisfies*

$$\text{cond}(\widehat{S}_D^{-1} S) \leq C \frac{(1 + \log n)^2}{\beta_n},$$

where C is independent of n and N .

The mixed elasticity case is more complicated, but an analogous wire basket preconditioner can be constructed and analyzed; see [45].

b) *A Neumann-Neumann preconditioner for Stokes problems.* In the Stokes case, we could also use a Laplacian-based Neumann-Neumann preconditioner on each scalar component; see Dryja and Widlund [22], Le Tallec [35] for a detailed analysis of this family of preconditioners for h -version finite elements and Pavarino [42] for an extension to spectral elements. In this case,

$$(22) \quad \widehat{S}_\Gamma = \begin{bmatrix} \widehat{S}_{NN} & 0 & 0 \\ 0 & \widehat{S}_{NN} & 0 \\ 0 & 0 & \widehat{S}_{NN} \end{bmatrix},$$

where

$$\widehat{S}_{NN}^{-1} = R_H^T K_H^{-1} R_H + \sum_{j=1}^N R_{\partial\Omega_j}^T D_j^{-1} \widehat{S}_j^\dagger D_j^{-1} R_{\partial\Omega_j}$$

is an additive preconditioner with independent coarse solver K_H^{-1} and local solvers \widehat{S}_j^\dagger , respectively associated with the coarse triangulation determined by the elements and with the boundary $\partial\Omega_j$ of each element. Here $R_{\partial\Omega_j}$ are restriction matrices returning the degrees of freedom associated with the boundary of Ω_j , D_j are diagonal matrices and \dagger denotes an appropriate pseudo-inverse for the singular Schur complements associated with interior elements; see [22, 42] for more details. Also for this preconditioner, a polylogarithmic bound is proven in [45].

THEOREM 3. *Let the blocks of the block-diagonal preconditioner \widehat{S}_D be the Neumann-Neumann preconditioner \widehat{S}_Γ defined in (22) and the coarse mass matrix C_0 . Then the Stokes saddle point Schur complement S preconditioned by \widehat{S}_D satisfies*

$$\text{cond}(\widehat{S}_D^{-1}S) \leq C \frac{(1 + \log n)^2}{\beta_n},$$

where C is independent of n and N .

Other scalar iterative substructuring preconditioners could also be applied in this fashion to the Stokes system; see Dryja, Smith, and Widlund [19].

8. Numerical results

In this section, we report the results of numerical experiments with the overlapping additive Schwarz method described in Section 6 and with some of the iterative substructuring methods described in Section 7. The two sets of results cannot be directly compared because the overlapping method is applied to h -version discretizations in two dimensions, while the iterative substructuring methods are applied to spectral element discretizations in three dimensions. All the computations were performed in MATLAB.

8.1. Overlapping Schwarz methods for h -version discretizations in two dimensions. In the following tables, we report the iteration counts for the iterative solution of our three model saddle point problems (Stokes, mixed elasticity, and Oseen) with the overlapping additive Schwarz method of Section 6, i.e. with the preconditioner \widehat{K}_{OAS}^{-1} defined in (10). In each application of our preconditioner, we solve the local and coarse saddle point problems directly by gaussian elimination. Inexact local and/or coarse solvers could also be considered, as in positive definite problems. We accelerate the iteration with GMRES, with zero initial guess and stopping criterion $\|r_i\|_2/\|r_0\|_2 \leq 10^{-6}$, where r_i is the i -th residual. Other Krylov space accelerators, such as BiCGSTAB or QMR, could be used. The computational domain Ω is the unit square, subdivided into $\sqrt{N} \times \sqrt{N}$ square subdomains. More complete results for Stokes problems, including multiplicative and other variants of the preconditioner, can be found in Klawonn and Pavarino [33].

a) $P_1(h) - P_1(2h)$ finite elements for the Stokes problem. Table 1 reports the iteration counts (with and without coarse solver) and relative errors in comparison with the direct solution (in the max norm) for the Stokes problem (1) discretized with $P_1(h) - P_1(2h)$ finite elements. Here $u = 0$ on $\partial\Omega$ and f is a uniformly distributed random vector. The overlap δ is kept constant and minimal, i.e. the size $\delta = 2h$ of one pressure element. h is refined and N is increased so that the subdomain size is kept constant at $H/h = 8$ (scaled speedup). The global problem size varies from 531 to 14,163 unknowns. The empty entry in the table (-) could not be run due to memory limitations. The results indicate that the number of iterations required by the algorithm is bounded by a constant independent of h and N . As in the positive definite case, the coarse problem is essential for scalability: without the coarse problem, the number of iterations grows with N . These results are also plotted in Figure 2 (left). The convergence history of GMRES (with and without \widehat{K}_{OAS}^{-1} as preconditioner) is shown in Figure 3 (left), for the case with 16 subdomains and $h = 1/32$.

TABLE 1. Stokes problem with $P_1(h) - P_1(2h)$ finite elements: iteration counts and relative errors for GMRES with the overlapping additive Schwarz preconditioner \hat{K}_{OAS}^{-1} ; constant subdomain size $H/h = 8$, minimal overlap $\delta = 2h$.

\sqrt{N}	h^{-1}	with coarse		no coarse	
		iter.	$\ x^m - x\ _\infty / \ x\ _\infty$	iter.	$\ x^m - x\ _\infty / \ x\ _\infty$
2	16	17	3.42e-7	21	9.05e-7
3	24	18	1.04e-6	33	1.82e-6
4	32	19	5.72e-7	43	9.53e-6
5	40	19	1.84e-6	53	1.07e-5
6	48	19	1.75e-6	63	1.39e-5
7	56	20	1.10e-6	73	3.17e-5
8	64	20	1.42e-6	86	4.60e-5
9	72	20	1.13e-6	-	-
10	80	20	1.79e-6	-	-

TABLE 2. Lid-driven cavity Stokes flow with $Q_1(h) - P_0(h)$ stab. finite elements: iteration counts and relative errors for GMRES with the overlapping additive Schwarz preconditioner \hat{K}_{OAS}^{-1} ; constant subdomain size $H/h = 8$.

overlap	\sqrt{N}	h^{-1}	with coarse		no coarse	
			iter.	$\ x^m - x\ _\infty / \ x\ _\infty$	iter.	$\ x^m - x\ _\infty / \ x\ _\infty$
$\delta = h$	2	16	18	5.58e-7	14	4.22e-1
	4	32	27	2.04e-6	27	4.88e-1
	8	64	31	7.35e-7	44	5.14e-1
$\delta = 2h$	2	16	16	1.93e-6	16	1.20e-7
	4	32	21	3.84e-7	37	8.06e-7
	8	64	22	2.51e-7	81	1.96e-6

b) $Q_1(h) - P_0(h)$ stabilized finite elements for the Stokes problem. Table 2 reports the iteration counts and relative errors in comparison with the direct solution for the Stokes problem (1) discretized with $Q_1(h) - P_0(h)$ stabilized finite elements, using the MATLAB software of Elman, Silvester, and Wathen [26], which requires $1/h$ to be a power of two. Here the boundary conditions and right-hand side are imposed to obtain a lid-driven cavity Stokes flow. The default value of the stabilization parameter β in (5) is $1/4$. The global problem size varies from 834 to 12,546 unknowns and, as before, we study the scaled speedup of the algorithm with $H/h = 8$. We could run only three cases ($N = 4, 16, 64$), but the iteration counts seem to behave as in the corresponding cases in Table 1 for $P_1(h) - P_1(2h)$ finite elements. Therefore the experiments seem to indicate a constant bound on the number of iterations that is independent of h and N . Again, the coarse space is essential for obtaining scalability. Here we can use a minimal overlap of $\delta = h$ since both velocities and pressures use the same mesh τ_h . We also report the results for $\delta = 2h$ to allow a comparison with the results of Table 1 (where the minimal

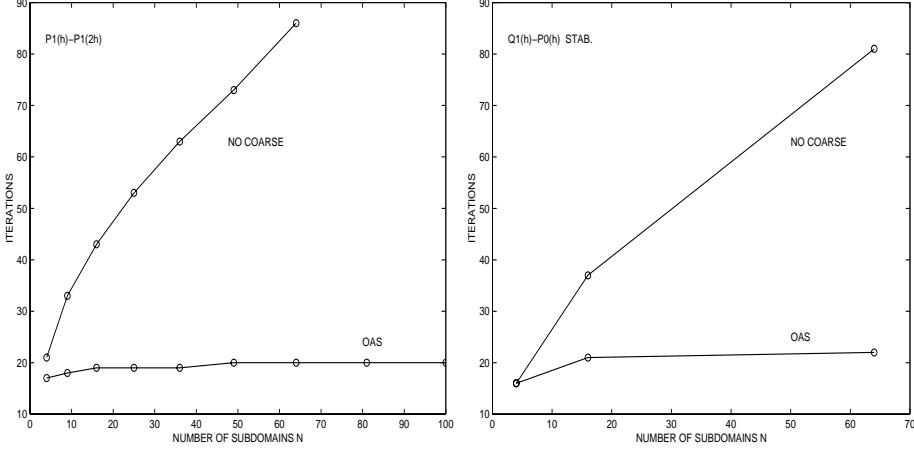


FIGURE 2. Iteration counts for GMRES with overlapping additive Schwarz preconditioner \hat{K}_{OAS}^{-1} (with and without coarse problem): subdomain size $H/h = 8$, overlap $\delta = 2h$, Stokes problem with $P_1(h) - P_1(2h)$ finite elements (left), lid-driven cavity Stokes flow with $Q_1(h) - P_0(h)$ stab. elements (right).

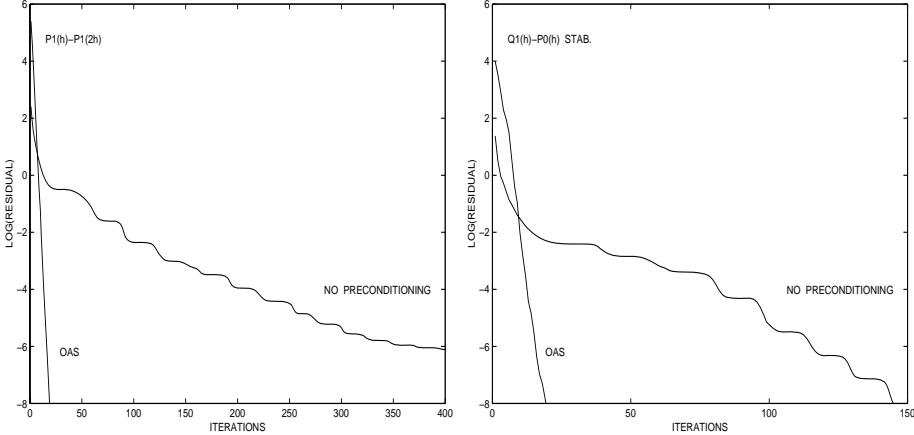


FIGURE 3. Convergence history for GMRES with and without overlapping additive Schwarz preconditioner \hat{K}_{OAS}^{-1} : $N = 16$, subdomain size $H/h = 8$, overlap $\delta = 2h$, Stokes problem with $P_1(h) - P_1(2h)$ finite elements (left), lid-driven cavity Stokes flow with $Q_1(h) - P_0(h)$ stab. elements (right).

overlap is $\delta = 2h$). These results are also plotted in Figure 2 (right). The convergence history of GMRES (with and without \hat{K}_{OAS}^{-1} as preconditioner) for the case $N = 16, h = 1/32, \delta = 2h$, is plotted in Figure 3 (right).

c) $P_1(h) - P_1(2h)$ finite elements for mixed elasticity. Analogous results were obtained for the mixed formulation of the elasticity system (2), discretized with $P_1(h) - P_1(2h)$ finite elements, with $u = 0$ on $\partial\Omega$. The results of Table 3 indicate

TABLE 3. Mixed linear elasticity with $P_1(h) - P_1(2h)$ finite elements: iteration counts for GMRES with the overlapping additive Schwarz preconditioner \widehat{K}_{OAS}^{-1} ; subdomain size $H/h = 8$, minimal overlap $\delta = 2h$.

\sqrt{N}	$1/h$	Poisson ratio ν						
		0.3	0.4	0.49	0.499	0.4999	0.49999	0.5
2	16	15	15	17	17	17	17	17
3	24	17	17	18	18	18	18	18
4	32	18	18	19	19	19	19	19
5	40	18	18	19	19	19	19	19
6	48	18	18	19	19	19	19	19
7	56	19	19	19	20	20	20	20
8	64	19	19	20	20	20	20	20
9	72	19	19	20	20	20	20	20
10	80	19	19	20	20	20	20	20

TABLE 4. Oseen problem with $Q_1(h) - P_0(h)$ stabilized finite elements and circular vortex $\mathbf{w} = (2y(1-x^2), -2x(1-y^2))$: iteration counts and relative errors for GMRES with the overlapping additive Schwarz preconditioner \widehat{K}_{OAS}^{-1} , constant subdomain size $H/h = 8$, overlap $\delta = h$.

	\sqrt{N}	h^{-1}	with coarse		no coarse	
			iter.	err.	iter.	err.
$\mu = 1$	2	16	19	1.12e-6	14	7.08e-7
	4	32	25	7.28e-7	31	2.71e-6
	8	64	30	7.86e-7	79	1.15e-6
$\mu = 0.1$	2	16	21	3.81e-7	15	5.47e-7
	4	32	26	6.03e-7	32	7.37e-7
	8	64	27	9.89e-7	99	3.27e-6
$\mu = 0.02$	2	16	29	9.43e-7	22	9.16e-7
	4	32	39	4.84e-7	42	4.81e-7
	8	64	42	1.15e-6	118	1.69e-6
$\mu = 0.01$	2	16	35	9.34e-7	29	9.53e-7
	4	32	51	2.02e-6	53	1.80e-6
	8	64	58	1.62e-6	211	1.45e-5

that the convergence rate of our method is bounded independently of h, N , and the Poisson ratio when approaching the incompressible limit $\nu = 0.5$.

d) $Q_1(h) - P_0(h)$ stabilized finite elements for the Oseen problem. Table 4 reports the iteration counts for GMRES with \widehat{K}_{OAS}^{-1} and $\delta = h$, and the relative errors in comparison with the direct solution, for the Oseen problem (3), using the MATLAB software of Elman, Silvester, and Wathen [26]. The divergence-free field \mathbf{w} is a circular vortex, $\mathbf{w} = (2y(1-x^2), -2x(1-y^2))$, and the stabilization parameter is 1/4. We study the scaled speedup of the algorithm with $H/h = 8$, running the three cases $N = 4, 16, 64$ for each given value of the diffusion parameter

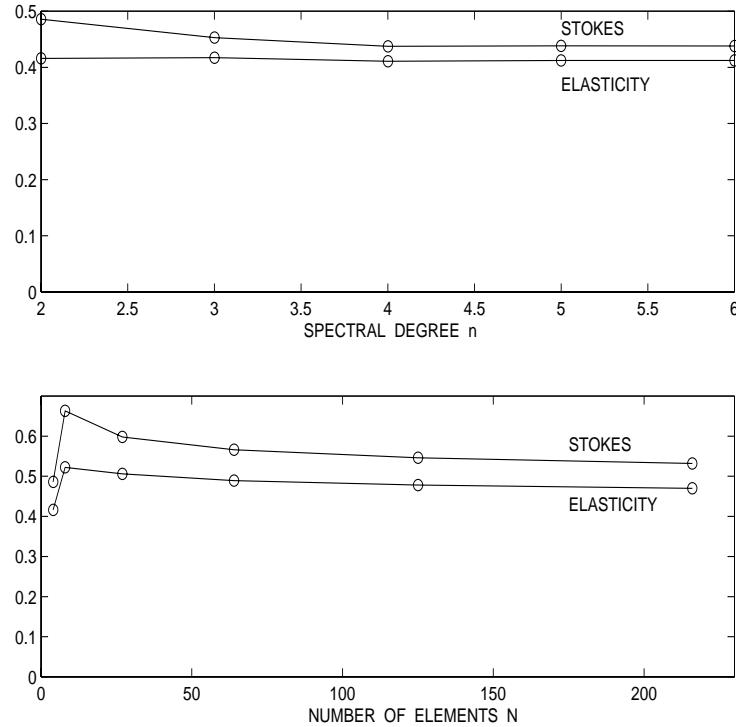


FIGURE 4. Inf-sup constant β_T for the Stokes and incompressible mixed elasticity saddle point Schur complement ($Q_n - Q_{n-2}$ spectral elements)

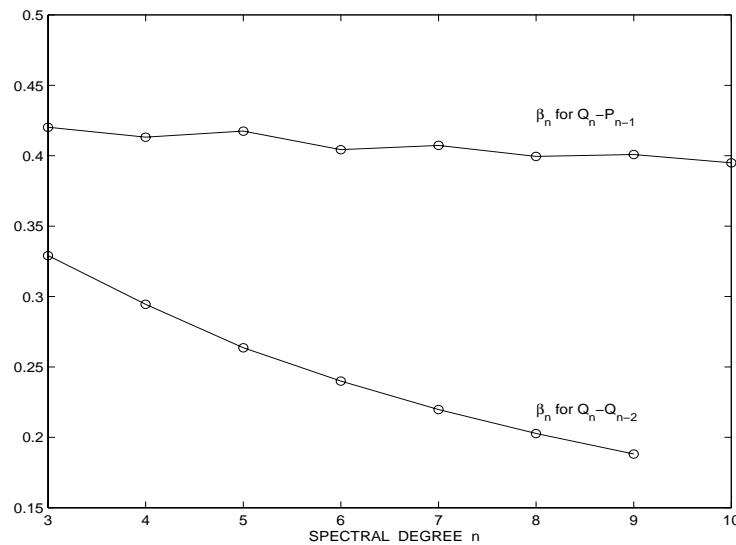


FIGURE 5. Inf-sup constant β_n for the discrete Stokes problem ($Q_n - Q_{n-2}$ and $Q_n - P_{n-1}$ spectral elements)

TABLE 5. Linear elasticity in mixed form: local condition number $\text{cond}(\widehat{S}_\Gamma^{-1} S_\Gamma)$ of the local saddle point Schur complement with wire basket preconditioner (with original wire basket block) on one interior element; $Q_n - Q_{n-2}$ method.

n	Poisson ratio ν						
	0.3	0.4	0.49	0.499	0.4999	0.49999	0.5
2	9.06	9.06	9.06	9.06	9.06	9.06	9.06
3	17.54	20.19	44.92	58.26	60.12	60.31	60.33
4	24.45	29.69	62.30	85.35	88.77	89.13	89.17
5	34.44	38.68	76.69	106.72	111.49	111.99	112.05
6	40.97	46.84	90.97	129.73	136.38	137.09	137.17
7	51.23	55.65	107.19	153.29	161.97	162.90	162.99
8	59.70	64.60	122.13	176.32	187.45	188.66	188.66

TABLE 6. Generalized Stokes problem: local condition number $\text{cond}(\widehat{S}_\Gamma^{-1} S_\Gamma)$ of the local saddle point Schur complement with wire basket preconditioner (with original wire basket block) on one interior element; $Q_n - Q_{n-2}$ method.

n	Poisson ratio ν						
	0.3	0.4	0.49	0.499	0.4999	0.49999	0.5
2	4.89	4.89	4.89	4.89	4.89	4.89	4.89
3	14.13	17.31	36.55	44.79	45.88	45.99	46.00
4	19.18	24.24	54.33	73.08	75.76	76.04	76.07
5	24.18	30.56	66.25	86.85	89.92	90.24	90.28
6	28.71	36.29	87.52	121.36	126.52	127.07	127.13
7	33.44	42.15	95.50	130.82	136.25	136.82	136.89
8	38.36	48.71	114.89	163.55	171.49	172.34	172.43

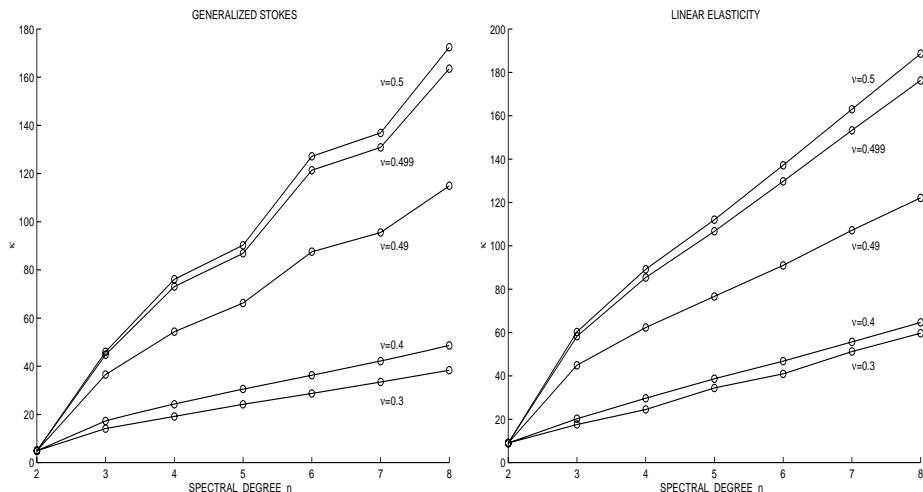


FIGURE 6. Local condition number $\text{cond}(\widehat{S}_\Gamma^{-1} S_\Gamma)$ from Tables 5 and 6; generalized Stokes problem (left), mixed elasticity (right)

μ . The results indicate a bound on the number of iterations that is independent of h and N , but that grows with the inverse of the diffusion parameter μ .

8.2. Iterative substructuring for spectral element discretizations in three dimensions. We first computed the discrete inf-sup constant β_Γ of the saddle point Schur complement (20), for both the mixed elasticity and Stokes system discretized with $Q_n - Q_{n-2}$ spectral elements. β_Γ is computed as the square root of the minimum nonzero eigenvalue of $C_0^{-1}B_0^T S_\Gamma^{-1} B_0$, where S_Γ and B_0 are the blocks in (20) and C_0 is the coarse pressure mass matrix. The upper plot in Figure 4 shows β_Γ as a function of the spectral degree n while keeping fixed a small number of elements, $N = 2 \times 2 \times 1$. The lower plot in Figure 4 shows β_Γ as a function of the number of spectral elements N for a small fixed spectral degree $n = 2$. Both figures indicate that β_Γ is bounded by a constant independent of N and n , in agreement with Lemma 1. We also computed the discrete inf-sup constant β_n of the whole Stokes problem on the reference cube by computing the square root of the minimum nonzero eigenvalue of $C_n^{-1}B_n^T A_n^{-1}B_n$, where A_n , B_n , and C_n are the blocks in (9). The results are plotted in Figure 5. The inf-sup parameter of the $Q_n - P_{n-1}$ method is much better than that of the $Q_n - Q_{n-2}$ method, in agreement with the theoretical results of [5] and the experiments in [43].

We next report on the local condition numbers of $\widehat{S}_\Gamma^{-1} S_\Gamma$ for one interior element. Here S_Γ is the velocity block in the saddle point Schur complement (20) and \widehat{S}_Γ^{-1} is the wire basket preconditioner described in Section 7 for the Stokes case. We report only the results obtained with the original wire basket block of the preconditioner, while we refer to Pavarino and Widlund [45] for more complete results. Table 5 presents the results for the mixed elasticity problem, while Table 6 gives the results for the generalized Stokes problem (in which there is a penalty term of the form $-t^2(p, q)_{L^2}$). These results are also plotted in Figure 6. In both cases, the incompressible limit is clearly the hardest, yielding condition numbers three or four times as large as those of the corresponding compressible case. For a given value of ν , the condition number seems to grow linearly with n , which is consistent with our theoretical results in Theorem 2 and 3.

References

1. K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, Stanford University Press, Stanfond, CA, 1958.
2. I. Babuška and M. Suri, *Locking effects in the finite element approximation of elasticity problems*, Numer. Math. **62** (1992), 439–463.
3. C. Bernardi and Y. Maday, *Approximations spectrales de problèmes aux limites elliptiques*, Springer-Verlag France, Paris, 1992.
4. ———, *Spectral Methods*, Handbook of Numerical Analysis, Volume V: Techniques of Scientific Computing (Part 2), North-Holland, 1997, pp. 209–485.
5. ———, *Uniform inf-sup conditions for the spectral element discretization of the Stokes problem*, Tech. Report 97034, Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie – Centre National de la Recherche Scientifique, October 1997.
6. D. Braess, *Stability of saddle point problems with penalty*, RAIRO M²AN **30** (1996), no. 6, 731–742.
7. D. Braess and C. Blömer, *A multigrid method for a parameter dependent problem in solid mechanics*, Numer. Math. **57** (1990), 747–761.
8. J.H. Bramble and J.E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp. **50** (1988), 1–17.
9. ———, *A domain decomposition technique for Stokes problems*, Appl. Numer. Math. **6** (1989/90), 251–261.

10. J.H. Bramble, J.E. Pasciak, and A. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal. **34** (1997), no. 3, 1072–1092.
11. S.C. Brenner, *Multigrid methods for parameter dependent problems*, RAIRO M²AN **30** (1996), no. 3, 265–297.
12. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, Berlin, 1991.
13. X.-C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Statist. Comput. **13** (1992), no. 1, 243–258.
14. ———, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems*, SIAM J. Numer. Anal. **30** (1993), no. 4, 936–952.
15. C. Canuto, *Stabilization of spectral methods by finite element bubble functions*, Comp. Meths. Appl. Mech. Eng. **116** (1994), no. 1–4, 13–26.
16. C. Canuto and V. Van Kemenade, *Bubble-stabilized spectral methods for the incompressible Navier-Stokes equations*, Comp. Meths. Appl. Mech. Eng. **135** (1996), no. 1–2, 35–61.
17. M. A. Casarin, *Schwarz preconditioners for spectral and mortar finite element methods with applications to incompressible fluids*, Ph.D. thesis, Dept. of Mathematics, Courant Institute of Mathematical Sciences, New York University, March 1996.
18. T. F. Chan and T. P. Mathew, Domain decomposition algorithms, Acta Numerica (1994), 61–143, Acta Numerica (1994), Cambridge University Press, 1994, pp. 61–143.
19. M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1662–1694.
20. M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.
21. ———, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), no. 3, 604–620.
22. ———, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math. **48** (1995), no. 2, 121–155.
23. H. C. Elman, *Preconditioning for the steady-state Navier-Stokes equations with low viscosity*, Tech. Report CS-TR-3712, UMIACS-TR-96-82, University of Maryland, 1996.
24. H. C. Elman and G. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1645–1661.
25. H. C. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations*, SIAM J. Sci. Comp. **17** (1996), 33–46.
26. H. C. Elman, D. J. Silvester, and A. J. Wathen, *Iterative methods for problems in computational fluid dynamics*, Iterative Methods in Scientific Computing (T. Chan R. Chan and G. Golub, eds.), Springer-Verlag, 1997.
27. R. E. Ewing and J. Wang, *Analysis of the Schwarz algorithm for mixed finite element methods*, Math. Model. Anal. Numer. **26** (1992), 739–756.
28. P.F. Fischer and E. Rønquist, *Spectral element methods for large scale parallel Navier-Stokes calculations*, Comp. Meths. Appl. Mech. Eng. **116** (1994), 69–76.
29. M. Fortin and D. Aboulaich, *Schwarz's decomposition method for incompressible flow problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds.), SIAM, 1988.
30. N. Kechkar and D. J. Silvester, *Analysis of locally stabilised mixed finite element methods for the Stokes problem*, Math. Comp. **58** (1992), 1–10.
31. A. Klawonn, *An optimal preconditioner for a class of saddle point problems with a penalty term*, Tech. report, Westfälische Wilhelms-Universität Münster, 1994, To appear in SIAM J. Sci. Comp., Vol. 19, #2 (1998).
32. ———, *Block-triangular preconditioners for saddle point problems with a penalty term*, Tech. report, Westfälische Wilhelms-Universität Münster, 1995, To appear in SIAM J. Sci. Comp., Vol. 19, #1 (1998).
33. A. Klawonn and L. F. Pavarino, *Overlapping Schwarz methods for mixed linear elasticity and Stokes problems*, Tech. Report 15-97 N, Westfälische Wilhelms-Universität Münster, 1997, To appear in Comp. Meths. Appl. Mech. Eng.

34. A. Klawonn and G. Starke, *Block Triangular Preconditioners for Nonsymmetric Saddle Point Problems: Field-of-Values Analysis*, Tech. Report 04/97-N, Westfälische Wilhelms-Universität Münster, Germany, Schriftenreihe Angewandte Mathematik und Informatik, March 1997.
35. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances (J. Tinsley Oden, ed.), vol. 1 (2), North-Holland, 1994, pp. 121–220.
36. P. Le Tallec and A. Patra, *Non-overlapping domain decomposition methods for adaptive hp approximations of the Stokes problem with discontinuous pressure fields*, Comp. Meths. Appl. Mech. Eng. **145** (1997), 361–379.
37. P. L. Lions, *On the Schwarz alternating method. I*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds.), SIAM, 1988.
38. Y. Maday, D. Meiron, A. Patera, and E. Rønquist, *Analysis of iterative methods for the steady and unsteady Stokes problem: application to spectral element discretizations*, SIAM J. Sci. Comp. **14** (1993), no. 2, 310–337.
39. Y. Maday, A. Patera, and E. Rønquist, *The $P_N \times P_{N-2}$ method for the approximation of the Stokes problem*, Tech. Report 92009, Dept. of Mech. Engr., M.I.T., 1992.
40. T. P. Mathew, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part I: Algorithms and Numerical results*, Numer. Math. **65** (1993), no. 4, 445–468.
41. ———, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part II: Theory*, Numer. Math. **65** (1993), no. 4, 469–492.
42. L. F. Pavarino, *Neumann-Neumann algorithms for spectral elements in three dimensions*, RAIRO M²AN **31** (1997), no. 4, 471–493.
43. ———, *Preconditioned mixed spectral element methods for elasticity and Stokes problems*, SIAM J. Sci. Comp. (1998), To appear.
44. L. F. Pavarino and O. B. Widlund, *A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions*, SIAM J. Numer. Anal. **33** (1996), no. 4, 1303–1335.
45. ———, *Iterative substructuring methods for spectral element discretizations of elliptic systems. II: Mixed methods for linear elasticity and Stokes flow*, Tech. Report 755, Dept. of Computer Science, Courant Institute of Mathematical Sciences, December 1997.
46. ———, *Iterative substructuring methods for spectral elements: Problems in three dimensions based on numerical quadrature*, Comp. Math. Appl. **33** (1997), no. 1/2, 193–209.
47. A. Quarteroni, *Domain decomposition algorithms for the Stokes equations*, Domain Decomposition Methods (Philadelphia) (T. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1989.
48. E. Rønquist, *A domain decomposition solver for the steady Navier-Stokes equations*, Proc. of ICOSAHOM '95 (A.V. Ilin and L.R. Scott, eds.), 1996.
49. T. Rusten, P. S. Vassilevski, and R. Winther, *Interior penalty preconditioners for mixed finite element approximations of elliptic problems*, Math. Comp. **65** (1996), no. 214, 447–466.
50. T. Rusten and R. Winther, *A preconditioned iterative method for saddle point problems*, SIAM J. Matr. Anal. Appl. **13** (1992), 887–904.
51. D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems. Part II: Using general block preconditioners*, SIAM J. Numer. Anal. **31** (1994), no. 5, 1352–1367.
52. B. F. Smith, P. Bjørstad, and W. D. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
53. M. Suri and R. Stenberg, *Mixed hp finite element methods for problems in elasticity and Stokes flow*, Numer. Math. **72** (1996), no. 3, 367–390.
54. R. Verfürth, *A multilevel algorithm for mixed problems*, SIAM J. Numer. Anal. **21** (1984), 264–271.
55. G. Wittum, *Multigrid methods for the Stokes and Navier-Stokes equations*, Numer. Math. **54** (1989), 543–564.

DEPARTMENT OF MATHEMATICS, UNIVERSITÀ DI PAVIA, VIA ABBIATEGRASSO 209, 27100
PAVIA, ITALY

E-mail address: pavarino@dragon.ian.pv.cnr.it

Parallel Implementation of Direct Solution Strategies for the Coarse Grid Solvers in 2-level FETI Method

François-Xavier Roux and Charbel Farhat

1. Introduction

The FETI method is based on introducing Lagrange multipliers along interfaces between subdomain to enforce continuity of local solutions [4]. It has been demonstrated to be numerically scalable in the case of second-order problems, thanks to a built-in “coarse grid” projection [3].

For high-order problems, especially with plate or shell finite element models in structural analysis, a two-level preconditioning technique for the FETI method has been introduced [2]. Computing the preconditioned gradient requires the solution of a coarse grid problem that is of the same kind as the original FETI problem, but is associated with a small subset of Lagrange multipliers enforcing continuity at cross-points. This preconditioner gives optimal convergence property for plate or shell finite element models [5].

This approach has been recently generalized to various local or partial continuity requirements in order to derive a general methodology for building second-level preconditioners [1].

For a sake of simplicity, the first method advocated for solving the coarse grid problems in distributed memory environment has been the same projected gradient as for the first-level FETI method [6] [7]. But with the increased complexity of the generalized 2-level FETI method, this approach leads to poor performance on machines with high performance compute nodes.

In the present paper this new preconditioning technique is reinterpreted in a simple algebraic form, in order to derive algorithms based on direct solution techniques to solve efficiently the coarse grid problems in distributed memory environment. Performance results for real-life applications are given.

2. Notations

In each subdomain, Ω_i , the local displacement field is solution of the linear elasticity equations with imposed forces on the interfaces with other subdomains:

$$(1) \quad K_i u_i = B_i^t \lambda + b_i$$

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65Y05.

Key words and phrases. Parallel Algorithms, Multi-Level Preconditioners.

where K_i is the stiffness matrix, u_i the displacement field, B_i a signed boolean matrix associated with the discrete trace operator, and λ the Lagrange multiplier, equal to the interaction forces between subdomains.

The continuity requirement along the interfaces is written as follows:

$$(2) \quad \sum_i B_i u_i = 0$$

where the signed discrete trace matrices B_i are such that if subdomains Ω_i and Ω_j are connected by the interface Γ_{ij} , then restriction of equation (2) on Γ_{ij} is: $u_i - u_j = 0$.

If the boundary of subdomain Ω_i does not contain a part of the external boundary with prescribed displacements, the local problem has only Neumann boundary conditions and then the matrix K_i is positive semi-definite.

If K_i^+ is a pseudo-inverse of matrix K_i , and if columns of matrix R_i form a basis of the kernel of K_i (rigid body motions), equation (1) is equivalent to:

$$(3) \quad \begin{cases} u_i = K_i^+(b_i + B_i^t \lambda) + R_i \alpha_i \\ R_i^t(b_i + B_i^t \lambda) = 0 \end{cases}$$

Introducing u_i given by equation (3) in the continuity condition (2) gives:

$$(4) \quad \sum_i B_i K_i^+ B_i^t \lambda + \sum_i B_i R_i \alpha_i = - \sum_i B_i K_i^+ b_i$$

With the constraint on λ set by the second equation of (3), the global interface problem can be written:

$$(5) \quad \begin{bmatrix} F & G \\ G^t & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix}$$

With:

- $F = \sum_i B_i K_i^+ B_i^t$, dual Schur complement matrix,
- $G\alpha = \sum_i B_i R_i \alpha_i$, jump of rigid body motions defined by α_i in Ω_i ,
- $(G^t \lambda)_i = R_i^t B_i^t \lambda$,
- $d = - \sum_i B_i K_i^+ b_i$,
- $c_i = -R_i^t b_i$.

In the following sections, we shall use the term “rigid body modes” for Lagrange multipliers in the image space of G .

3. Parallelization of the rigid body projection

3.1. Rigid body projection. Thanks to the fact that the number of admissibility constraints related to the second set of equations:

$$(6) \quad (G^t \lambda)_i = R_i^t B_i^t \lambda$$

the hybrid condensed system (5) can be solved in practice by a projected gradient algorithm. The projection associated with the rigid body modes can be explicitly computed:

$$(7) \quad P = I - G(G^t G)^{-1} G^t$$

The computation of the product by projection P requires products by G and G^t and the solution of systems with form:

$$(8) \quad (G^t G)\alpha = G^t g$$

The product by G^t can be performed independently in each subdomain, the product by G requires exchanging data through interfaces between neighboring subdomains. The product by $(G^t G)^{-1}$ requires the solution of a coarse grid problem associated with rigid body motions coefficients in each subdomain. This problem has the same kind of algebraic structure as a finite element problem whose elements are the subdomains and whose degrees of freedom in each element are the subdomain rigid body motions coefficients.

3.2. Forming and factorization of $(G^t G)$. The $(G^t G)$ matrix has a sparse block structure. If subscripts i and j are associated with subdomains Ω_i and Ω_j , the block $(G^t G)_{ij}$ representing entries in $(G^t G)$ associated with influence between rigid body modes in Ω_i and Ω_j is equal to:

$$(9) \quad (G^t G)_{ij} = R_i^t B_i^t B_j R_j = (B_i R_i)^t (B_j R_j)$$

The columns of $B_i R_i$ and $B_j R_j$ are respectively the traces on interfaces of rigid body motions of subdomains Ω_i and Ω_j . The entries of these columns are simultaneously non zero only on interface Γ_{ij} . So, the computation of block $(G^t G)_{ij}$ just requires the values of the traces on Γ_{ij} of rigid body motions of subdomains Ω_i and Ω_j .

In order to minimize computation costs and memory requirements, the following algorithm can be implemented to compute and factorize the $(G^t G)$ matrix.

1. Store in each subdomain Ω_i , the traces of rigid body motions on each interface Γ_{ij} with neighboring subdomain Ω_j . This requires the storage for each interface Γ_{ij} of a matrix $(B_i R_i)_j$ with number of rows equal to the number of degrees of freedom on interface Γ_{ij} and number of columns equal to the number of rigid body motions in Ω_i .
2. Exchange $(B_i R_i)_j$ matrices with neighboring subdomains. This means that subdomain Ω_i sends matrix $(B_i R_i)_j$ to subdomain Ω_j and receives from it matrix $(B_j R_j)_i$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{ij} and number of columns equal to the number of rigid body motions in Ω_j .
3. In subdomain Ω_i , compute the following matrix-matrix products for each interface Γ_{ij} :

$$(10) \quad \begin{aligned} (G^t G)_{ij} &= (B_i R_i)_j^t (B_j R_j)_i \\ (G^t G)_{ii} &= (G^t G)_{ii} + (B_i R_i)_j^t (B_i R_i)_j \end{aligned}$$

Now, subdomain Ω_i has a part of the sparse block structure of matrix $(G^t G)$.

4. Assemble the complete $(G^t G)$ matrix via global data transfer operations (“GATHER”).
5. Factorize the complete $(G^t G)$ matrix via Choleski factorization using optimal renumbering strategy. Note that numerical pivoting strategy may also be implemented to detect global rigid body motions in the case of a not clamped global problem. In this case, the global rigid boy motions form the kernel of the $(G^t G)$ matrix.
6. Compute the rows of the $(G^t G)^{-1}$ matrix associated to the rigid body modes in the neighborhood of subdomain, including the subdomain itself and its neighbors (see next section).

3.3. Computation of rigid body modes projection. The projected gradient is given by:

$$(11) \quad Pg = g - G(G^t G)^{-1} G^t g$$

The computation of $G^t g$ is purely local. In subdomain Ω_i , $(G^t g)_i = R_i^t B_i^t g$.

The main step of the projection is the product by $(G^t G)^{-1}$. As the global $(G^t G)$ matrix has been factorized, the product by $(G^t G)^{-1}$ requires a forward-backward substitution. To perform it, the global $G^t g$ vector must be assembled.

Once $\alpha = (G^t G)^{-1} G^t g$ has been computed a product by G must be performed to compute the projected gradient. As $G\alpha$ is defined as:

$$(12) \quad G\alpha = \sum B_i R_i \alpha_i$$

its restriction on interface Γ_{ij} is given by:

$$(13) \quad (G\alpha)_{ij} = (B_i R_i)_j \alpha_i + (B_j R_j)_i \alpha_j$$

This means that subdomain Ω_i can compute the projected gradient on its interfaces without any data transfer, provided that it has the values of α in all its neighboring subdomains. Furthermore, it does not need at all the values of α in the other subdomains.

Hence, the computation of the projection can be parallelized with minimal data transfer, provided that each subdomain computes the solution of the coarse problem:

$$(14) \quad (G^t G)\alpha = G^t g$$

for its neighborhood including the subdomain itself and its neighbors. Only the rows of the $(G^t G)^{-1}$ matrix associated to the neighborhood are required in the subdomain to do so. These rows form a matrix with number of rows equal to the number of rigid body motions in the neighborhood and number of columns equal to the total number of rigid body modes. Thanks to the symmetry of $(G^t G)$, the rows of this matrix are equal to the columns of $(G^t G)^{-1}$ associated to the neighborhood. So, computing this matrix noted $(G^t G)_{zone_i}^{-1}$ requires a forward-backward substitution with complete matrix $(G^t G)$ for each row. The computation of this matrix represents the last step of the forming and factorization of $(G^t G)$, as indicated in the previous section.

So, computing the projected gradient in parallel requires the following steps:

1. Compute $(G^t g)_i = R_i^t B_i^t g$ in each subdomain Ω_i .
2. Gather complete $\beta = G^t g$ in each subdomain via a global data transfer operation.
3. Compute components of $\alpha = (G^t G)^{-1} \beta$ in neighborhood of each subdomain Ω_i . This means compute the matrix-vector product:

$$(15) \quad (G^t G)_{zone_i}^{-1} \beta$$

4. Compute $G\alpha$ in each subdomain Ω_i as:

$$(16) \quad (G\alpha)_{ij} = (B_i R_i)_j \alpha_i + (B_j R_j)_i \alpha_j$$

for each interface Γ_{ij} .

5. Compute $Pg = g - G\alpha$ in each subdomain.

The only transfer this algorithm requires is the gathering of $\beta = G^t g$ in step 2.

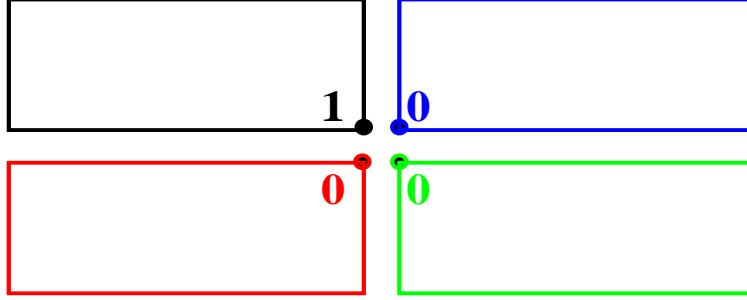


FIGURE 1. A “corner motion” for a scalar problem

4. The second level FETI preconditioner

4.1. Definition of corner modes. In this section, the second level FETI preconditioner is presented in the case of a coarse grid defined as the so-called “corner modes”. The objective consists in constraining the Lagrange multiplier to generate local displacement fields that are continuous at interface cross-points. To get a practical formulation of this constraint, it can be observed that requiring the continuity of displacement fields at interface cross-points is equivalent to imposing their jump to be orthogonal to the jump of “corner motions” defined as displacement fields with unit value in one space direction at a node connected to a crosspoint as in Figure (1).

Note C_i the set of corner motions in subdomain Ω_i , then the Lagrange multiplier λ satisfies the continuity requirement of associated displacement fields at interface cross-points if the projected gradient satisfies:

$$(17) \quad (B_i C_i)^t P g = 0 \quad \forall i \Leftrightarrow \left(\sum_i B_i C_i \gamma_i \right)^t P g = 0 \quad \forall \gamma$$

4.2. Coarse grid space. Let us define “corner modes” from corner motions and the associated global operator C in the same way as rigid body modes and operator G are defined from rigid body motions:

$$(18) \quad C\gamma = \sum_i B_i C_i \gamma_i$$

The second-level FETI preconditioner consists in building the search direction vector w from the projection of the gradient Pg in order to satisfy the constraint of generating local displacement fields that are continuous at interface cross-points. As the jump of displacement fields created by w is equal to PFw , this constraint can be written in the following way:

$$(19) \quad C^t PFw = C^t P^t Fw = 0$$

In order to satisfy this constraint, the search direction vector w must be constructed from Pg corrected by a vector in the image space of C :

$$(20) \quad w = Pg + C\gamma$$

The search direction vector w must also satisfy the constraint of orthogonality to the traces of rigid body modes. This constraint can be written $Pw = w$. So w must

have the following form:

$$(21) \quad w = Pg + C\gamma + G\beta \text{ with } Pw = w$$

So, the coarse grid space associated with the second-level FETI preconditioner must contain both image spaces of C and G . This means that it is the image space of the matrix noted $[CG]$ whose first columns are the columns of C and last columns the columns of G .

4.3. Second-level FETI problem . By definition of the rigid body modes projection P , PFw satisfies:

$$(22) \quad G^t PFw = G^t P^t Fw = 0$$

From the definition of the coarse grid space (21), the search direction vector must have the following form:

$$(23) \quad w = Pw = Pg + P(C\gamma + G\beta)$$

The formulation of constraints in equations (19) and (22), entails that γ and β satisfy the following problem:

$$(24) \quad \begin{aligned} C^t P^t FP(C\gamma + G\beta) &= -C^t P^t FPg \\ G^t P^t FP(C\gamma + G\beta) &= -G^t P^t FPg \end{aligned}$$

These equations can be rewritten as:

$$(25) \quad [CG]^t P^t FP [CG] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -[CG]^t P^t FPg$$

This system is precisely a coarse FETI problem, posed in the subspace of Lagrange multipliers defined as the image space of $[CG]$. With this coarse grid preconditioner, the solution algorithm appears clearly as a two-level FETI method: at each iteration of projected conjugate gradient at the fine level, an additional preconditioning problem of the same type has to be solved at the coarse grid level.

4.4. Rigid body projection for coarse grid vectors. The rigid body projection takes a simple form for vectors belonging to the coarse grid space. In general, for any Lagrange multiplier μ , the rigid body projection is defined by:

$$(26) \quad P\mu = \mu + G\delta \text{ with } G^t(\mu + G\delta) = 0$$

In the same way, for a vector in the coarse grid space, the projection can be written:

$$(27) \quad P(C\gamma + G\beta) = C\gamma + G\beta + G\delta = C\gamma + G(\beta + \delta)$$

correction δ being defined by the constraint:

$$(28) \quad G^t(C\gamma + G(\beta + \delta)) = 0$$

So, $(\beta + \delta)$ satisfies:

$$(29) \quad (G^t G)(\beta + \delta) = -G^t C\gamma$$

Hence, the projection of a vector in the coarse grid space is given by the following equation:

$$(30) \quad P(C\gamma + G\beta) = C\gamma - G(G^t G)^{-1} G^t C\gamma$$

This equation illustrates the fact that the effective degrees of freedom of the second-level FETI preconditioner are the corner modes coefficients. The rigid body modes have been added to the coarse grid space just for allowing coarse grid vectors to

satisfy the rigid body constraint. For any set of corner modes coefficients γ , the only coarse grid vector satisfying the rigid body constraint is given by equation (30).

Let us note R_{GC} the matrix defined by:

$$(31) \quad R_{GC} = -(G^t G)^{-1} G^t C$$

$R_{GC}\gamma$ represents the coefficients of the rigid body modes correction to apply on the corner mode generated by coefficients γ for satisfying the rigid body constraint.

Note P_{coarse} the following matrix:

$$(32) \quad P_{coarse} = \begin{bmatrix} I \\ R_{GC} \end{bmatrix} [I \ 0]$$

Equation (30) can be rewritten:

$$(33) \quad P [CG] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = [CG] P_{coarse} \begin{bmatrix} \gamma \\ \beta \end{bmatrix}$$

So, actually P_{coarse} is the rigid body projection for coarse grid vectors in the natural coarse grid basis defined by columns of $[CG]$.

4.5. Projected coarse grid problem. The second-level FETI problem (25) can be rewritten as:

$$(34) \quad (P [CG])^t F (P [CG]) \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -(F P [CG])^t Pg$$

Thanks to the definition of the rigid body projection for coarse grid vectors given by equation (33), a new formulation of the second-level FETI problem can be derived from equation (34):

$$(35) \quad P_{coarse}^t ([CG]^t F [CG]) P_{coarse} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -P_{coarse}^t (F [CG])^t Pg$$

Note F_{coarse} the projection of the FETI operator in the coarse grid space:

$$(36) \quad F_{coarse} = [CG]^t F [CG]$$

From the definition of P_{coarse} in equation (32), the second-level FETI problem can be rewritten:

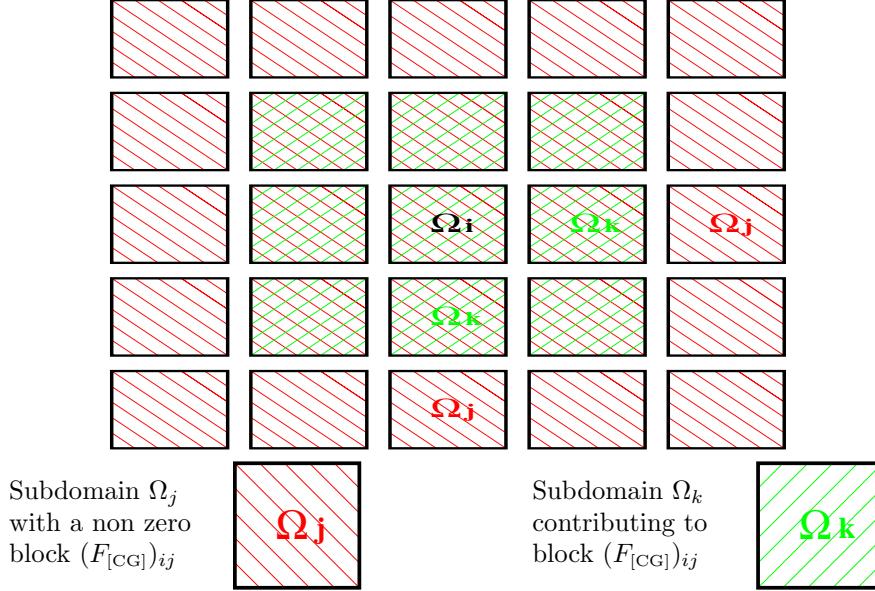
$$(37) \quad \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{GC} \end{bmatrix} [I \ 0] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -\begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t (F [CG])^t Pg$$

Thanks to the definition of the rigid body projection for coarse grid vectors, the only degrees of freedom of equation (37) are coefficients of γ . Eliminating null blocks in this equation finally leads to the projected coarse grid problem that defines γ :

$$(38) \quad \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{GC} \end{bmatrix} \gamma = -\begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t (F [CG])^t Pg$$

From equation (23) and (30), the search direction vector satisfying both corner modes constraint and rigid body modes constraint is given by:

$$(39) \quad w = Pw = Pg + P(C\gamma + G\beta) = Pg + [CG] \begin{bmatrix} I \\ R_{GC} \end{bmatrix} \gamma$$

FIGURE 2. Contribution of subdomains to blocks of F_{coarse}

5. Parallel computation of the projected coarse grid FETI matrix

5.1. Forming of F_{coarse} . The projection of FETI operator on the coarse grid space is defined in equation (36). The coarse grid matrix $[CG]$ and the first-level FETI matrix F can be written:

$$(40) \quad \begin{aligned} [CG] &= \sum_j B_j [C_j R_j] \\ F &= \sum_k B_k K_k^+ B_k^t \end{aligned}$$

where $[C_j R_j]$ is the matrix whose columns are the corner and rigid body motions of subdomain Ω_j .

From this equation, it can be derived that F_{coarse} has a sparse block structure, such that the block of interaction between coarse grid degrees of freedom in subdomains Ω_i and Ω_j is defined as:

$$(41) \quad (F_{coarse})_{ij} = (B_i [C_i R_i])^t \left(\sum_k B_k K_k^+ B_k^t \right) (B_j [C_j R_j])$$

As a consequence of the fact that the trace operator B_n is non zero only on interfaces of subdomain Ω_n , a block of form:

$$(42) \quad (B_i [C_i R_i])^t (B_k K_k^+ B_k^t) B_j [C_j R_j]$$

is non zero only if subdomain Ω_k is neighbor of both subdomains Ω_i and Ω_j . Such a block is the contribution of subdomain Ω_k to the block matrix $(F_{coarse})_{ij}$. This means that the contribution of subdomain Ω_k to the coarse-grid FETI matrix F_{coarse} is a dense square matrix with dimension equal to the sum of numbers of corner and rigid body motions in its neighborhood, including the subdomain itself and all its neighbors. Figure (2) show the dependency between subdomains for the computation of $(F_{coarse})_{ij}$.

5.2. Local contribution to F_{coarse} . From equation (42), it appears that subdomain Ω_k can compute its contribution to the F_{coarse} matrix, if it has all the traces of corner and rigid body motions in its neighborhood. So, forming the contribution of a subdomain to the F_{coarse} matrix can be organized according to the algorithm described below.

1. Store in each subdomain Ω_k , the traces of corner motions on each interface Γ_{kj} with neighboring subdomain Ω_j . This requires the storage for each interface Γ_{kj} of a matrix $(B_k C_k)_j$ with number of rows equal to the number of degrees of freedom on interface Γ_{kj} and number of columns equal to the number of corner motions in Ω_k .
2. Exchange $(B_k C_k)_j$ matrices with neighboring subdomains. This means that subdomain Ω_k sends matrix $(B_k C_k)_j$ to subdomain Ω_j and receives from it matrix $(B_j C_j)_k$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{kj} and number of columns equal to the number of corner motions in Ω_j .
3. In subdomain Ω_k , perform a forward-backward substitution for each local corner and rigid body motion, and for each corner and rigid body motion of neighboring subdomains, in order to compute the following matrices:

$$(43) \quad K_k^+ B_k^t B_j C_j \text{ and } K_k^+ B_k^t B_j R_j$$

for $j = k$ or j such that Ω_j is a neighbor of Ω_k .

4. Store the traces of resulting vectors interface by interface. On each interface Γ_{ki} , these traces form two sets of matrices $(B_k K_k^+ B_k^t B_j C_j)_i$ and $(B_k K_k^+ B_k^t B_j R_j)_i$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{ki} and number of columns respectively equal to the number of corner or rigid body motions in Ω_j
5. In subdomain Ω_k , for each interface Γ_{ki} , compute the following matrix-matrix products:

$$(44) \quad \begin{aligned} & (B_i C_i)_k^t (B_k K_k^+ B_k^t B_j C_j)_i && (B_i C_i)_k^t (B_k K_k^+ B_k^t B_j R_j)_i \\ & (B_i R_i)_k^t (B_k K_k^+ B_k^t B_j C_j)_i && (B_i R_i)_k^t (B_k K_k^+ B_k^t B_j R_j)_i \end{aligned}$$

In the same way, the following blocks must be computed and added to the contribution of other interfaces:

$$(45) \quad \begin{aligned} & (B_k C_k)_i^t (B_k K_k^+ B_k^t B_j C_j)_i && (B_i C_i)_k^t (B_k K_k^+ B_k^t B_j R_j)_i \\ & (B_k R_k)_i^t (B_k K_k^+ B_k^t B_j C_j)_i && (B_i R_i)_k^t (B_k K_k^+ B_k^t B_j R_j)_i \end{aligned}$$

Now, subdomain Ω_k has its contribution to the $(F_{coarse})_{ij}$ block, for any pair of its neighboring subdomains Ω_i and Ω_j .

5.3. Contribution of subdomain to the projected coarse grid FETI problem. If the restriction of coarse grid on neighborhood of subdomain Ω_k , including Ω_k itself, is written $I^{(k)}$, then the projected coarse grid problem of equation (38) is constructed from the local contributions to the coarse grid FETI matrix as follows:

$$(46) \quad \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{GC} \end{bmatrix} = \sum_k \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t (I^{(k)})^t F_{coarse}^{(k)} (I^{(k)}) \begin{bmatrix} I \\ R_{GC} \end{bmatrix}$$

where $F_{coarse}^{(k)}$ is the contribution of Ω_k to the coarse grid FETI matrix constructed in the way presented in section 5.2.

So, once $F_{coarse}^{(k)}$ has been formed, the computation of the contribution of subdomain Ω_k to the projected coarse grid FETI matrix only requires the rows of R_{GC} associated with rigid body modes in the neighborhood of Ω_k .

In the same way, once the projected coarse grid problem of equation (38) is solved, the computation of the vector w as described in equation (39) can be performed without any data transfer in each subdomain, provided that the complete solution γ and the rows of R_{GC} associated with rigid body modes in the neighborhood are known.

5.4. Forming of the rigid body projection for coarse grid vectors. From the previous section, it appears that each subdomain must construct the rows of the rigid body projection associated with the rigid body modes of its neighborhood. In subdomain Ω_k , these rows form the following matrix:

$$(47) \quad I^{(k)} \begin{bmatrix} I \\ R_{GC} \end{bmatrix} = I^{(k)} \begin{bmatrix} I \\ -(G^t G)^{-1} G^t C \end{bmatrix}$$

The $G^t C$ matrix has exactly the same kind of sparse block structure as $G^t G$ and can be formed in parallel using the same methodology. The $(G^t G)$ matrix has already been formed and factorized in each subdomain. The only problem to compute the rows of R_{GC} associated with the rigid body modes in the neighborhood of subdomain Ω_k lies in the fact that all the entries corresponding to corner motions in ALL subdomains must be computed.

The following algorithm can be implemented.

1. In subdomain Ω_k , compute the following matrix-matrix products for each interface Γ_{kj} :

$$(48) \quad \begin{aligned} (G^t C)_{jk} &= (B_j R_j)_k^t (B_k C_k)_j \\ (G^t C)_{kk} &= (G^t C)_{kk} - (B_j R_j)_k^t (B_k C_k)_j \end{aligned}$$

The result is scattered in a matrix whose number of columns is equal to the number of corner motions in subdomain Ω_k and number of rows equal to the total number of rigid body modes. This matrix forms the subset of columns of $G^t C$ associated with corner modes of subdomain Ω_k .

2. Compute $-(G^t G)^{-1} G^t C$ for all columns of $G^t C$ associated with corner modes of subdomain Ω_k . This requires a number of forward-backward substitutions, using the COMPLETE factorization of $(G^t G)$, equal to the number of corner motions in subdomain Ω_k . Now, subdomain Ω_k has all the columns of

$$(49) \quad R_{GC} = -(G^t G)^{-1} G^t C$$

associated with its own corner modes.

3. Exchange the columns of R_{GC} associated with corner modes of subdomain Ω_k with neighboring subdomains. Now, subdomain Ω_k has all the columns of R_{GC} associated with all corner modes in its neighborhood, itself included.
4. Subdomain Ω_k has all the COLUMNS of R_{GC} associated with its own corner modes. To get the ROWS of R_{GC} associated with its rigid body modes, a global matrix transposition through data exchange between all processors must be performed.
5. A last data transfer operation with neighboring subdomains must be performed in order to get rows of R_{GC} associated with rigid body modes in the whole neighborhood of each subdomain.

6. Build the matrix:

$$(50) \quad I^{(k)} \begin{bmatrix} I \\ R_{GC} \end{bmatrix}$$

The rows of R_{GC} associated with rigid body modes in the whole neighborhood of each subdomain form the lower part of this matrix, the upper part is just the boolean restriction to the subset of corner modes in the whole neighborhood of each subdomain.

5.5. Forming and factorization of the projected coarse grid FETI matrix . The forming and factorization of the complete projected coarse grid FETI matrix can be performed in 4 steps.

1. In each subdomain Ω_k , compute the matrix-matrix product:

$$(51) \quad \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t (I^{(k)})^t F_{coarse}^{(k)} (I^{(k)}) \begin{bmatrix} I \\ R_{GC} \end{bmatrix}$$

The result is a square matrix of dimension equal to the total number of corner modes. It is the contribution of subdomain Ω_k to the projected coarse grid FETI matrix.

2. Assemble the projected coarse grid FETI matrix through a global data transfer. This is a reduction with add operation.
3. Build the pseudo-inverse of the projected coarse grid FETI matrix via Choleski factorization with numerical pivoting. This matrix is not full rank when there is some redundancy of the corner modes.
4. Compute the rows of the pseudo-inverse of the projected coarse grid FETI associated with the corner modes in the neighborhood of subdomain, including the subdomain itself and its neighbors.

6. Parallel solution of the second-level FETI problem

6.1. Right-hand side of the projected coarse grid problem. From equation (38) it can be derived that the first step of the computation of the right-hand side of the projected coarse grid problem requires the computation of:

$$(52) \quad (F [CG])^t Pg = ((\sum_k B_k K_k^+ B_k^t) (\sum_j B_j [C_j R_j]))^t Pg$$

All the data to compute the contribution of subdomain Ω_k to this product are already present, thanks to the computation of the local contribution to F_{coarse} , as explained in section 5.2. Also, the columns of the rigid body projection for the coarse grid problem have been computed locally.

Hence, the parallelization of the computation of the right-hand side of the projected coarse grid problem can be performed as follows:

1. In subdomain Ω_k , for each interface Γ_{ki} , compute the following matrix-vector products:

$$(53) \quad \begin{aligned} & (B_k K_k^+ B_k^t B_j C_j)_i^t (Pg)_{ki} \\ & (B_k K_k^+ B_k^t B_j R_j)_i^t (Pg)_{ki} \end{aligned}$$

for all subdomains Ω_j in the neighborhood of Ω_k , itself included.

Scatter the resulting vectors in a vector of dimension equal to the sum of corner and rigid body motions in all subdomains in the locations associated with the corner and rigid body modes of the neighborhood of Ω_k . If $j = k$,

the contributions of all interfaces must be added. This vector represents the contribution of subdomain Ω_k to $(F[CG])^t Pg$.

2. Assemble $(F[CG])^t Pg$ via global data exchange. This is a reduction with add operation.
3. Compute in each subdomain Ω_k the part of the product:

$$(54) \quad - \begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t (F[CG])^t Pg$$

associated with corner modes of Ω_k . Only the columns of the matrix R_{GC} associated with corner modes of Ω_k and that have been computed during the forming of the rigid body projection for the coarse grid, are needed.

This procedure gives the restriction to each subdomain of the right-hand side of the projected coarse grid problem.

4. Gather complete right-hand side of the projected coarse grid problem in each subdomain via a global data transfer operation.

6.2. Solution of the projected coarse grid FETI problem. Once the complete right-hand side has been gathered in each subdomain, the computation of the solution γ of the projected coarse grid problem (38) in subdomain Ω_i just requires a product by the matrix:

$$(55) \quad \left(\begin{bmatrix} I \\ R_{GC} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{GC} \end{bmatrix} \right)_{zone_i}^{-1}$$

using the rows of the pseudo-inverse of the projected coarse grid FETI associated with the corner modes in the neighborhood of subdomain computed as explained in section 5.5.

6.3. Computation of the search direction vector. From equation (39) it can be observed that the search direction vector w can be computed locally in each subdomain, provided that entries of the solution of the projected coarse grid FETI problem γ and of vector $\beta = R_{GC}\gamma$ associated to the subdomain and its neighbors are known.

The procedure described in the previous section has given the entries of the solution of the projected coarse grid FETI problem γ associated to the subdomain and its neighbors. The computation of the search direction vector w can be completed as follows.

1. In each subdomain, compute the contribution to β by computing the matrix-vector product:

$$(56) \quad \beta = R_{GC}\gamma$$

for columns of R_{GC} associated with corner modes of subdomain.

2. Assemble β through a global data transfer. This is a reduction with add operation. Extract the entries of β associated to the subdomain and its neighbors.
3. In subdomain Ω_i , for each interface Γ_{ij} , compute the following matrix-vector products:

$$(57) \quad w_{ij} = Pg_{ij} + (B_i C_i)_j \gamma_i + (B_j C_j)_i \gamma_j + (B_i R_i)_j \beta_i + (B_j R_j)_i \beta_j$$

This step computes without any data transfer the restriction to subdomain of:

$$(58) \quad w = Pg + C\gamma + G\beta = Pg + [CG] \begin{bmatrix} I \\ R_{GC} \end{bmatrix} \gamma$$

6.4. Computation of the starting λ . For a given λ^0 satisfying the rigid body constraint:

$$(59) \quad (G^t \lambda^0)_i = -R_i^t b_i$$

in each subdomain Ω_i , note g^0 the gradient $F\lambda^0 - d$. The corner mode correction of λ^0 , w , must be of form:

$$(60) \quad w = P(C\gamma + G\beta)$$

The corrected initial λ , is $\lambda^0 + w$, and the associated corrected gradient is $g^0 + Fw$. The correction w must be such that the projected corrected gradient satisfy:

$$(61) \quad \begin{aligned} C^t P(g^0 + Fw) &= 0 \\ G^t P(g^0 + Fw) &= 0 \end{aligned}$$

From these equations, the same development as in section 4.3 leads to the second level FETI problem:

$$(62) \quad [CG]^t P^t F P [CG] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -[CG]^t P g^0$$

This problem is similar to the one of equation (25), except for the right-hand-side. As, by definition of the rigid body projection P , $G^t Pg^0$ is equal to 0, the right-hand side can be even simplified:

$$(63) \quad [CG]^t P^t F P [CG] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -C^t Pg^0$$

This problem can be solved in the same way as for the search direction vector, except for the right-hand side that is simpler and can be computed exactly in the same way as the right-hand side of a rigid body projection, using corner motions C_i instead of rigid body motions G_i .

7. Application

7.1. “Interface averaging” modes. The second-level preconditioner has been presented in the previous sections for the case of a coarse grid associated with “corner modes”. This preconditioner has been demonstrated to be very efficient for dealing with the singularity at cross-points for high order problems like plate and shell finite element problems. In this case, the coarse grid is built for imposing a local continuity requirement. But the same approach can be used to enforce more global continuity requirements, like for instance continuity of mean value of each component of the displacement field on each interface between two subdomains.

The corresponding modes, called interface averaging modes are simply the jumps of local constant motion on a single interface as in Figure 3. Once again, building the coarse grid space as the set of jumps of local special motions makes it simpler to define and gives automatically admissible Lagrange multipliers. This approach also allows to define efficient coarse grid preconditioner in the case where local Neumann problems are well posed, especially for time-dependent problems.

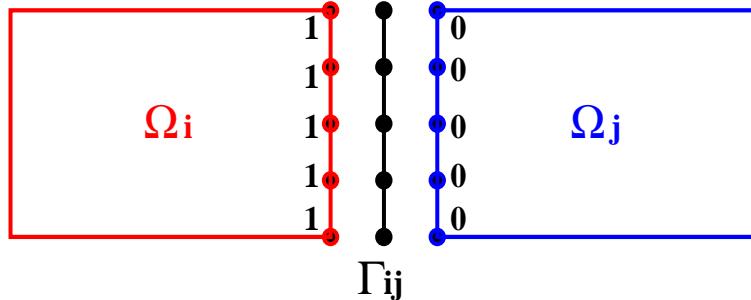


FIGURE 3. Generation of a single interface averaging mode for a scalar problem

TABLE 1. Solution time on IBM-SP2 with corner and interface averaging modes

Numb. of Domains	1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
16	52	21	26	11	19	8.4
32	91	14	28	5	20	3.5

In such a case, there is no rigid body projection to play the role of a first-level coarse grid preconditioner, and the standard FETI method is not numerically scalable.

7.2. Parallel performance. To illustrate the efficiency of the second-level FETI preconditioner parallelized by the solution technique presented in this paper, a small shell problem, with 25600 nodes and 6 degrees of freedom per node, has been solved on an IBM-SP2 system with either 16 or 32 processors and one domain per processor.

The reason why a small problem has been chosen is the following: the objective is to demonstrate that with the parallelization technique developed in this paper, the 2-level FETI method is not only numerically scalable, but its implementation on distributed memory systems actually gives scalable performances. The main drawback with the coarse grid preconditioners is the fact that their implementation on distributed memory machines can be very inefficient because the number of data transfers they require is high and their granularity is very low. Of course, for a given target architecture, if the local size of the problem is large enough, the cost for the forward-backward substitution may remain dominant. But to be really scalable, a method must be efficient even in the case where the size of the local problems is not very large.

The stopping criterion is the same in all cases presented in this paper and is related to the global residual:

$$(64) \quad \|Ku - b\| / \|b\| < 10^{-6}$$

Also, in all cases presented in this paper, the local optimal Dirichlet preconditioner is used. Table 1 gives a comparison of the elapsed parallel times for the iterations of 1-level and 2-level FETI methods with various coarse grid preconditioners. It

TABLE 2. Solution time on IBM-SP2 with corner and interface averaging modes

1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
163	106	25	11	16	7.8

TABLE 3. Number of modes and times for building the second-level preconditioner

1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
Numb. of Rigid Body Modes	Dirichlet + Neumann(s)	Numb. of Modes	Set-up Time(s)	Numb. of Modes	Set-up Time(s)
304	17	588	20	924	40

demonstrates clearly that the cost for the parallel solution of the coarse grid problems is small enough to ensure that the solution time decreases in the same proportion as the number of iterations. Enforcing the corner continuity is enough to make the method scalable for this kind of shell problem. Nevertheless, the averaging modes give a significant decrease of the number of iterations. Speed-ups with the 2-level FETI method are even super-linear, thanks to the decrease of the bandwidth of local matrices with larger number of subdomains.

7.3. Constructing cost for the coarse grid preconditioner. In the previous section, only the timings for the iterations were given. But the cost for assembling, factorizing and inverting the second-level FETI operator can be far from negligible in comparison with the time for the initial factorization of the local Dirichlet and Neumann problems matrices.

In order to illustrate this point, a larger shell model problem with 100000 nodes and 600000 degrees of freedom has been decompose in 64 subdomains. With such a large number of subdomains, the global number of coarse grid modes with both corner and averaging modes can be nearly one thousand. It would clearly not make sense to use such a large number of coarse grid modes for solving a small problem with only a few thousands degrees of freedom. Nevertheless, with 64 subdomains, the number of nodes per subdomain is less than 2000, hence the time for factorizing the local Dirichlet and Neumann matrices is rather small.

Table 2 features the number of iterations and the parallel times for the iterations of 1-level and 2-level FETI methods with various coarse grid preconditioners. Table 3 shows the total number of rigid body modes and of corner and corner + averaging modes, and it gives a comparison of times spent on each processor for factorizing the local Dirichlet and Neumann matrices and for assembling and factorizing the second-level FETI preconditioner.

In the case of the largest coarse grid space, this table shows that the time for forming the second-level preconditioner can be more than two times larger than the factorization time of local Dirichlet and Neumann matrices. Hence, this time may

be considered too high, especially in comparison with the time for the iterations shown in Table 2.

Nevertheless these results are quite good. First, as discussed above, the global size of this problem is not that large for such a number of subdomains. The time for local factorizations and forward-backward substitutions increases faster with the number of nodes per subdomain than the time for building the second-level FETI preconditioner. So, for larger problems, the comparison will be more in favor of the 2-level method.

Secondly, this time to build the second-level FETI preconditioner is payed only once in the case of multiple right-hand-side, and the overall efficiency will be even better in such a case.

Thirdly, for the timings presented here, the factorization of the second-level FETI matrix has been computed in sequential. It would be possible to use a parallel skyline method to perform this factorization that represents nearly half the time spent in the construction of the second-level FETI preconditioner. Even a very low efficiency would be enough to make the factorization time itself negligible.

8. Conclusion

Thanks to the algebraic interpretation of the 2-level FETI method presented here a parallel implementation methodology has been designed. It allows using global direct solvers for the second-level FETI preconditioner but keeps working with a simple description of the interfaces at subdomain level.

The actual efficiency obtained with this approach makes feasible the enrichment of the coarse grid spaces used in the preconditioner, making the overall method faster and more robust.

References

1. F. Risler C. Farhat, P. C. Chen and F.-X. Roux, *A simple and unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers*, Comput. Meths. Appl. Mech. Engrg. (in press).
2. C. Farhat and J. Mandel, *The two-level FETI method for static and dynamic plate problems-part1: an optimal iterative solver for biharmonic systems*, Comput. Meths. Appl. Mech. Engrg. (in press).
3. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Meths. Appl. Mech. Engrg. **115** (1994), 367–388.
4. C. Farhat and F.-X. Roux, Implicit parallel processing in structural mechanics (J. Tinsley Oden, ed.), Computational Mechanics Advances, vol. 2, Nort-Holland, 1994, pp. 1–124.
5. J. Mandel, R. Tezaur, and C. Farhat, *An optimal Lagrange multiplier based domain decomposition method for plate bending problems*, SIAM J. Sc. Stat. Comput. (in press).
6. F.-X. Roux, *Parallel implementation of a domain decomposition method for non-linear elasticity problems*, Domain-Based Parallelism and Problem decomposition Methods in Computational Science and Engineering (Philadelphia) (Youcef Saad David E. Keyes and Donald G. Truhlar, eds.), SIAM, 1995, pp. 161–176.
7. F.-X. Roux and C. Farhat, *Parallel implementation of the two-level FETI method*, Domain Decomposition Methods for Partial Differential Equations, 10th International Conference, Bergen, Norway, 1996 (M. Espedal P. Bjorstad and D. Keyes, eds.), Wiley and sons.

HIGH PERFORMANCE COMPUTING DPT, ONERA, BP72, F92322 CHATILLON CEDEX, FRANCE
E-mail address: roux@onera.fr

CENTER FOR SPACE STRUCTURES, UNIVERSITY OF COLORADO AT BOULDER, BOULDER CO
80309-0526
E-mail address: charbel@alexandra.colorado.edu

Domain Decomposition and Multi-Level Type Techniques for General Sparse Linear Systems

Yousef Saad, Maria Sosonkina, and Jun Zhang

1. Introduction

Domain-decomposition and multi-level techniques are often formulated for linear systems that arise from the solution of elliptic-type Partial Differential Equations. In this paper, generalizations of these techniques for irregularly structured sparse linear systems are considered. An interesting common approach used to derive successful preconditioners is to resort to Schur complements. In particular, we discuss a multi-level domain decomposition-type algorithm for iterative solution of large sparse linear systems based on independent subsets of nodes. We also discuss a Schur complement technique that utilizes incomplete LU factorizations of local matrices.

A recent trend in parallel preconditioning techniques for general sparse linear systems is to exploit ideas from domain decomposition concepts and develop methods which combine the benefits of superior robustness of ILU-type preconditioning techniques with those of scalability of multi-level preconditioners. Two techniques in this class are the Schur complement technique (Schur-ILU) developed in [19] and the point and block multi-elimination ILU preconditioners (ILUM, BILUM) discussed in [16, 21].

The framework of the Schur-ILU preconditioner is that of a distributed sparse linear system, in which equations are assigned to different processors according to a mapping determined by a graph partitioner. The matrix of the related Schur complement system is also regarded as a distributed object and never formed explicitly. The main difference between our approach and the methods described in [2, 7], is that we do not seek to compute an approximation to the Schur complement. Simply put, our Schur-ILU preconditioner is an approximate solve for the global system which is derived by solving iteratively the Schur complement equations corresponding to the interface variables. For many problems, it has been observed that the number of steps required for convergence remains about the same as the number of processors and the problem size increase. The overall solution time increases slightly, at a much lower rate than standard Schwarz methods.

1991 *Mathematics Subject Classification*. Primary 65F10; Secondary 65F50, 65N55, 65Y05.

This research was supported in part by NSF under grant CCR-9618827, and in part by the Minnesota Supercomputer Institute.

ILUM and BILUM exploit successive independent set orderings. One way to think of the idea underlying ILUM is that by a proper reordering of the original variables, the matrix is put in the form

$$(1) \quad A = \begin{pmatrix} B & F \\ E & C \end{pmatrix},$$

where B is diagonal so that the Schur complement system associated with the C block remains sparse. Then the idea is applied recursively, computing a sequence of Schur complement (or reduced) systems. The last of these reduced systems is solved by an iterative solver. This recursively constructed preconditioner has a multi-level structure and a good degree of parallelism. Similar preconditioners have been designed and tested in [4, 21] to show near grid-independent convergence for certain type of problems. In a recent report, some of these multi-level preconditioners have been tested and compared favorably with other preconditioned iterative methods and direct methods at least for the Laplace equation [3]. Other multi-level preconditioning and domain decomposition techniques have also been developed in finite element analysis or for unstructured meshes [1, 5].

The ILUM preconditioner has been extended to a block version (BILUM) in which the B block in (1) is block-diagonal. This method utilizes independent sets of small clusters (or blocks), instead of single nodes [4, 21]. In some difficult cases, the performance of this block version is substantially superior to that of the scalar version. The major difference between our approach and the approaches of Botta and Wubs [4] and Reusken [13] is in the choice of variables for the reduced system. In [4] and [13], the nodes in the reduced system, i.e., the unknowns associated with the submatrix C in (1), are those nodes of the independent set itself and this leads to a technique which is akin to an (algebraic) multigrid approach [14]. An approximate inverse technique is used to invert the top-left submatrix B in (1) which is no longer diagonal or block-diagonal. In their implementations, these authors employ a simple approximate inverse technique which usually requires diagonal dominance in the B matrix. In contrast, our approach chooses the nodes of the reduced system to be those unknowns associated with the complement to the independent set. Therefore the difference is that B is associated with the independent set instead of C . This approach is more akin to a domain decomposition technique and it is more generally applicable since it does not require diagonal dominance in either the B or C submatrix.

One aim of the current paper is to further extend BILUM techniques of [21] to include blocks of large size and to treat related issues of keeping sparsity of the BILUM factors. Measurable parameters are introduced to characterize the efficiency of a preconditioner. Numerical results with some hard-to-solve problems are presented to demonstrate the merits of the new implementations. For the Schur-LU preconditioner, we compare the performance of various options for solving the local problems and make some observations and recommendations.

2. Schur Complements and Recursive Schur Complements

Consider a linear system of the form

$$(2) \quad Ax = b,$$

where A is a large sparse nonsymmetric real matrix of size n . To solve such a system on a distributed memory computer, a graph partitioner is usually first invoked to

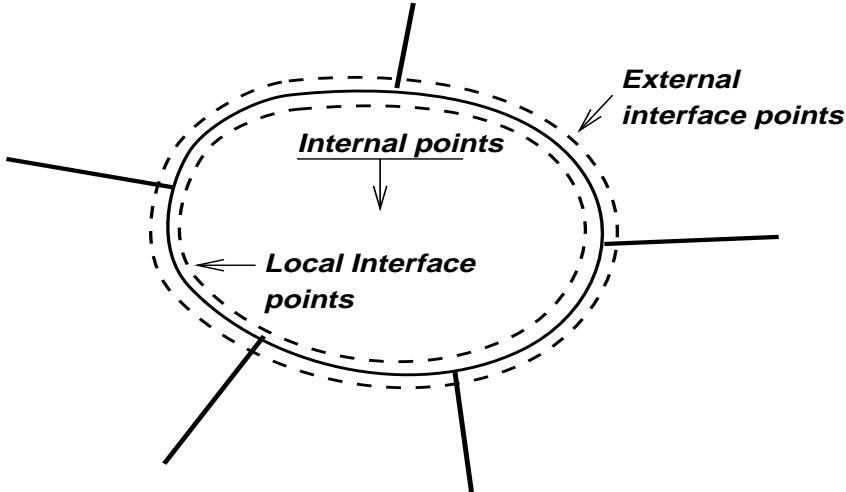


FIGURE 1. A local view of a distributed sparse matrix.

partition the adjacency graph of A . The data is then distributed to processors such that pairs of equations-unknowns are assigned to the same processor. When this is done, three types of unknowns can be distinguished. (1) Interior unknowns that are coupled only with local equations; (2) Local interface unknowns that are coupled with both non-local (external) and local equations; and (3) External interface unknowns that belong to other subdomains and are coupled with local equations. This setting which is illustrated in Figure 1, is common to most packages for parallel iterative solution methods [7, 9, 10, 11, 15, 18, 22, 23].

2.1. Distributed sparse linear systems. The matrix assigned to a certain processor is split into two parts: the *local* matrix A_i , which acts on the local variables and an *interface matrix* X_i , which acts on the external variables. Accordingly, the local equations can be written as follows:

$$(3) \quad A_i x_i + X_i y_{i,ext} = b_i,$$

where x_i represents the vector of local unknowns, $y_{i,ext}$ are the external interface variables, and b_i is the local part of the right-hand side vector. It is common to reorder the local equations in such a way that the interface points are listed last after the interior points. This ordering leads to an improved interprocessor communication and to reduced local indirect addressing during matrix-vector multiplication. Thus, the local variables form a local vector of unknowns x_i which is split into two parts: the subvector u_i of internal vector components followed by the subvector y_i of local interface vector components. The right-hand side b_i is conformally split into the subvectors f_i and g_i . When the block is partitioned according to this splitting, the local equations (3) can be written as follows:

$$(4) \quad \begin{pmatrix} B_i & F_i \\ E_i & C_i \end{pmatrix} \begin{pmatrix} u_i \\ y_i \end{pmatrix} + \begin{pmatrix} 0 \\ \sum_{j \in N_i} E_{ij} y_j \end{pmatrix} = \begin{pmatrix} f_i \\ g_i \end{pmatrix}.$$

Here, N_i is the set of indices for subdomains that are neighbors to the subdomain i . The term $E_{ij}y_j$ is a part of the product $X_iy_{i,ext}$ which reflects the contribution to the local equation from the neighboring subdomain j . These contributions are the result of multiplying X_i by the external interface unknowns:

$$\sum_{j \in N_i} E_{ij}y_j \equiv X_iy_{i,ext}.$$

The result of this multiplication affects only the local interface unknowns, which is indicated by a zero in the top part of the second term of the left-hand side of (4).

2.2. Schur complement systems. This section gives a brief background on Schur complement systems; see e.g., [17, 22], for additional details and references. The Schur complement system is obtained by eliminating the variable u_i from the system (4). Extracting from the first equation $u_i = B_i^{-1}(f_i - F_iy_i)$ yields, upon substitution in the second equation,

$$(5) \quad S_i y_i + \sum_{j \in N_i} E_{ij}y_j = g_i - E_i B_i^{-1} f_i \equiv g'_i,$$

where S_i is the “local” Schur complement

$$(6) \quad S_i = C_i - E_i B_i^{-1} F_i.$$

The equations (5) for all subdomains i ($i = 1, \dots, p$) constitute a system of equations involving only the interface unknown vectors y_i . This reduced system has a natural block structure related to the interface points in each subdomain:

$$(7) \quad \begin{pmatrix} S_1 & E_{12} & \dots & E_{1p} \\ E_{21} & S_2 & \dots & E_{2p} \\ \vdots & \ddots & \vdots & \vdots \\ E_{p1} & E_{p-1,2} & \dots & S_p \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} g'_1 \\ g'_2 \\ \vdots \\ g'_p \end{pmatrix}.$$

The diagonal blocks in this system, the matrices S_i , are dense in general. The off-diagonal blocks E_{ij} , which are identical with those involved in the global system (4), are sparse. The system (7), which we rewrite in the form

$$Sy = g',$$

is the Schur complement system and S is the “global” Schur complement matrix.

2.3. Induced global preconditioners. It is possible to develop preconditioners for the *global system* (2) by exploiting methods that *approximately solve the reduced system* (7). These techniques, termed “induced preconditioners” (see, e.g., [17]), are based on a reordered version of the global system (2) in which all the internal vector components $u = (u_1, \dots, u_p)^T$ are labeled first followed by all the interface vector components y ,

$$(8) \quad \left(\begin{array}{ccc|c} B_1 & & & F_1 \\ & B_2 & & F_2 \\ & & \ddots & \vdots \\ \hline E_1 & E_2 & \cdots & E_p \end{array} \middle| \begin{array}{c} F_1 \\ F_2 \\ \vdots \\ \hline C \end{array} \right) \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \\ y \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \\ g \end{pmatrix},$$

which also can be rewritten as

$$(9) \quad \begin{pmatrix} B & F \\ E & C \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}.$$

Note that the B block acts on the interior unknowns. Consider the block LU factorization

$$(10) \quad \begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \begin{pmatrix} B & F \\ 0 & S \end{pmatrix},$$

where S is the global Schur complement

$$S = C - EB^{-1}F.$$

This Schur complement matrix is identical to the coefficient matrix of system (7) (see, e.g., [17]). The global system (9) can be preconditioned by an approximate LU factorization constructed such that

$$(11) \quad L = \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} B & F \\ 0 & M_S \end{pmatrix}$$

with M_S being some approximation to S .

Note that the effect of the forward solve is to compute the modified right-hand side g' for the Schur complement system (7). Once this is done, the backward solve with the matrix U consists of two additional steps: solving with M_S to compute approximations to y , and then back-substituting to compute the approximations to the u variables.

Therefore, a global preconditioning operation induced by a Schur complement solve would consist of the following three steps.

1. Compute the reduced right-hand side $g' = g - EB^{-1}f$;
2. Approximately solve $M_S y = g'$;
3. Back-substitute for the u variables, i.e., solve $Bu = f - Fy$.

Each of the above three steps can be accomplished in different ways and this leads to a rich variety of options. For example, in (1) and (2) the linear system with B can be done either iteratively or directly, or approximately using just an ILU factorization for B . The choices for (2) are also numerous. One option considered in [19] starts by replacing (5) by an approximate system of the form,

$$(12) \quad y_i + \tilde{S}_i^{-1} \sum_{j \in N_i} E_{ij} y_j = \tilde{S}_i^{-1} [g_i - E_i B_i^{-1} f_i],$$

in which \tilde{S}_i is some (local) approximation to the local Schur complement matrix S_i . This can be viewed as a block-Jacobi preconditioned version of the Schur complement system (7) in which the diagonal blocks S_i are approximated by \tilde{S}_i . The above system is then solved by an iterative accelerator such as GMRES requiring a solve with \tilde{S}_i at each step. In one method considered in [19] the approximation \tilde{S}_i was extracted from an Incomplete LU factorization of A_i . The idea is based on the following observation (see [17]). Let A_i be the matrix on the left-hand side of (4) and assume it is factored as $A_i = L_i U_i$, where

$$(13) \quad L_i = \begin{pmatrix} L_{B_i} & 0 \\ E_i U_{B_i}^{-1} & L_{S_i} \end{pmatrix} \quad \text{and} \quad U_i = \begin{pmatrix} U_{B_i} & L_{B_i}^{-1} F_i \\ 0 & U_{S_i} \end{pmatrix}.$$

Then, $L_{S_i} U_{S_i}$ is equal to the Schur complement S_i associated with the partitioning (4), see [17, 19]. Thus, an approximate LU factorization of S_i can be obtained

canonically from an approximate factorization to A_i , by extracting the related parts from the L_i and U_i matrices.

2.4. Recursive Schur complements. The diagonal blocks S_i in the Schur complement system (7) are usually dense blocks. However, the off-diagonal terms E_{ij} are sparse. As a result, it is interesting to consider a particular situation when the blocks B_i are all of small size, e.g., of size one or two. In this situation the coefficient matrix in (7) remains sparse and it is natural to think about a recursive application of the induced preconditioning technique described above. The second level of reduction can be applied to Schur complement system (7) resulting in the second-level Schur complement system (“the Schur complement for the Schur complement”). This process can be continued for a few more levels and the last level system can be solved with a standard iterative method. This idea was exploited in [16] for blocks B_i of the smallest possible size, namely one. In [21], the idea was generalized to blocks larger than one and a number of heuristics were suggested for selecting these blocks.

We recall that an independent set is a set of unknowns which are not coupled by an equation. A maximal independent set is an independent set that cannot be augmented by other elements to form another independent set. These notions can be generalized by considering subsets of unknowns as a group. Thus, a block independent set is a set of such groups of unknowns such that there is no coupling between unknowns of any two different groups. Unknowns within the same group may be coupled.

ILUM and BILUM can be viewed as a recursive applications of domain decomposition in which the subdomains are all of small size. In ILUM the subdomains are all of size one and taken together they constitute an independent set. In BILUM [21], this idea was slightly generalized by using block-independent sets, with groups (blocks) of size two or more instead of just one. As the blocks (subdomains) become larger, Schur complements become denser. However, the resulting Schur complement systems are also smaller and they tend to be better conditioned as well.

3. Block Independent Sets with Large Blocks

Heuristics based on local optimization arguments were introduced in [21] to find Block Independent Sets (BIS) having various properties. It has been shown numerically that selecting new subsets according to the lowest possible number of outgoing edges in the subgraph, usually yields better performance and frequently the smallest reduced system. These algorithms were devised for small independent sets. Extending these heuristic algorithms for extracting Block Independent Sets with large block sizes is straightforward. However, these extensions may have some undesirable consequences. First, the cost is not linear with respect to the block size and it can become prohibitive as the block size increases. The second undesirable consequence is the rapid increase in the amount of fill-ins in the LU factors and in the inverse of the block diagonal submatrix. As a result, the construction and application of a BILUM preconditioner associated with a BIS having large subsets tend to be expensive [21].

Suppose a block independent set (BIS) with a uniform block size k has been found and the matrix A is permuted into a two-by-two block matrix of the form

(with the unknowns of the independent set listed first)

$$(14) \quad A \sim PAP^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix},$$

where P is a permutation matrix associated with the BIS ordering and $B = \text{diag}[B_1, B_2, \dots, B_s]$ is a block diagonal matrix of dimension $m = ks$, where s is the number of uniform blocks of size k . The matrix C is square and of dimension $l = n - m$. In [21], a block ILU factorization of the form (11) is performed, i.e.,

$$(15) \quad \begin{pmatrix} B & F \\ E & C \end{pmatrix} \approx \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \times \begin{pmatrix} B & F \\ 0 & A_1 \end{pmatrix} = L \times U.$$

Here $A_1 = C - EB^{-1}F$ is the Schur complement and I is the identity matrix. In order to maintain sparsity a dropping strategy is adopted when computing the submatrices EB^{-1} and A_1 , based on a threshold tolerance. The BILUM preconditioner is obtained by recursively applying the above procedures to these successive Schur complements up to a certain number of levels, say $nlev$. The last reduced system obtained is solved by a direct method or a preconditioned iterative method.

Once the BILUM preconditioner is constructed, the solution process (application of BILUM) consists of the (block) forward and backward steps [16]. At each step (level), we partition the vector x_j as

$$(16) \quad x_j = \begin{pmatrix} y_j \\ x_{j+1} \end{pmatrix}$$

corresponding to the two-by-two block matrix (14). The BILUM preconditioning amounts to performing the following steps:

Copy the right-hand side vector b to x_0 .

For $j = 0, 1, \dots, nlev - 1$, do forward sweep:

 Apply permutation P_j to x_j to partition it in the form (16).

$$x_{j+1} := x_{j+1} - E_j B_j^{-1} y_j.$$

End do.

Solve with a relative tolerance ε :

$$A_{nlev} x_{nlev} := x_{nlev}.$$

For $j = nlev - 1, \dots, 1, 0$, do backward sweep:

$$y_j := B_j^{-1} (y_j - F_j x_{j+1}).$$

 Apply inverse permutation P_j^T to the solution y_j .

End do.

BILUM is, in effect, a recursive application of a domain decomposition technique. In the successive Schur complement matrices obtained, each block contains the internal nodes of a subdomain. The inverse and application of all blocks on the same level can be done in parallel. One distinction with traditional domain decomposition methods [12, 22] is that all subdomains are constructed algebraically and exploit no physical information. In addition, the reduced system (coarse grid acceleration) is solved by a multi-level recursive process akin to a multigrid technique.

We define several measures to characterize the efficiency of BILUM (and other preconditioning techniques). The first one is called the *efficiency ratio* (e-ratio) which is defined as the ratio of the preprocessing time over iteration time, i.e., the ratio of the CPU time spent computing the BILUM preconditioner to that required by GMRES/BILUM to converge. The efficiency ratio determines how expensive it is to compute a preconditioner, relative to the time spent in the iteration phase. It

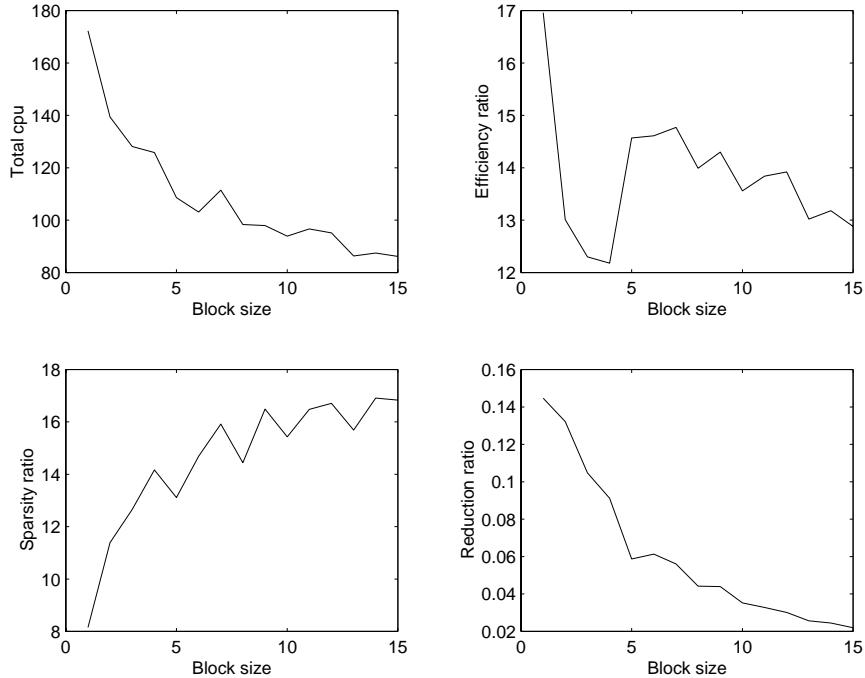


FIGURE 2. Characteristic measures for solving the 5-POINT matrix for different sizes of uniform blocks.

should be used with the second measure that is called the *total CPU* time which is the CPU time in seconds that a computer spends to compute the preconditioner and to solve the linear system. Given a total CPU time, a good preconditioner should not be too expensive to compute.

The third measure is the *sparsity ratio* (*s*-ratio) which is the ratio of the number of nonzeros of the BILUM factors to that of the matrix A . Note that the number of nonzeros for BILUM includes all the nonzeros of the LU factors at all levels plus those of the last reduced system and its preconditioner. If a direct method is used to solve the last reduced system, this latter number is to be replaced by the number of nonzero elements of the (exact) LU factorization of this system. The sparsity ratio determines how much memory is needed to store the given preconditioner, compared with that needed for the simplest preconditioner ILU(0). If the sparsity ratio is too large, a preconditioned iterative solver may lose one of its major advantages over a direct solver. The fourth measure is the *reduction ratio* (*r*-ratio) which is the ratio of the dimension of the last reduced system to that of the original system A . The reduction ratio determines how good an algorithm finds the independent set. The total CPU time, efficiency ratio and sparsity ratio may be suitable to characterize other preconditioning techniques, but the reduction ratio is mainly for the BILUM-type preconditioners. These four characteristic measures are more informative than the measure provided by the iteration count alone.

Our iterative algorithm consists of GMRES with a small restart value as an accelerator and BILUM as a preconditioner. The last reduced system is solved iteratively by another GMRES preconditioned by an ILUT preconditioner [17].

TABLE 1. Description of test matrices.

matrix	size	nonzeros	description
5-POINT	40 000	199 200	5-point upwind scheme of convection-diffusion
RAEFSKY3	21 200	1 488 768	Fluid structure interaction turbulence problem
VENKAT50	62 424	1 717 792	Unstructured 2D Euler solver, time step = 50
WIGTO966	3 864	238 252	Euler equation model

The dropping strategy that we used in [21] is the simplest one. Elements in EB^{-1} in the L factor and in the reduced system A_1 are dropped whenever their absolute values are less than a threshold tolerance $droptol$ times the average value of the current row. For BILUM with large size BIS formed by the greedy algorithm, this simple single dropping strategy is not sufficient to keep BILUM sparse enough. Figure 2 shows the behavior of the four characteristic measures as the block size changes when an algorithm using this single dropping strategy is used to solve a system with the 5-POINT matrix described in [21] (some information about this matrix is given in Table 1). Here, a 20-level BILUM with single dropping strategy was used. The coarsest level solve was preconditioned by ILUT($10^{-4}, 10$). The iteration counts are 5 for block sizes ≤ 4 , and 4 otherwise. It can be seen that although BILUM with large block sizes reduced all three other measures (not monotonically), it increased the storage cost substantially. The sparsity ratio was doubled when the block size increased from 1 to 15. Such an uncontrolled large storage requirement may cause serious problems in large scale applications.

Inspired by the dual threshold dropping strategy of ILUT [17], we propose a similar dual threshold dropping for BILUM. We first apply the single dropping strategy as above to the EB^{-1} and A_1 matrices and keep only the largest $lfil$ elements (absolute value) in each row.

Another cause of loss of sparsity comes from the matrix B^{-1} . In general, each block of B is sparse, but the inverse of the block is dense. For BIS with large blocks this results in a matrix B^{-1} that is much denser than B . However, if a block is diagonally dominant, the elements of the block inverse are expected to decay away from the diagonal rapidly. Hence, small elements of B^{-1} can be dropped without sacrificing the quality of the preconditioner too much. In practice, we may use a double dropping strategy similar to the one just suggested, for the EB^{-1} and A_1 matrices, possibly with different parameters.

4. Numerical Experiments

The Schur-ILU factorization has been implemented in the framework of the PSPARSLIB package [15, 18, 20] and was tested on a variety of machines. The experiments reported here have been performed on a CRAY T3E-900. ILUM and Block ILUM have been implemented and tested on sequential machines. Additional experiments with BILUM, specifically with small blocks, have been reported elsewhere, see [21]. We begin with experiments illustrating the Schur-ILU preconditioners.

Some information on the test matrices is given in Table 1. The Raefsky matrix was supplied to us by H. Simon from Lawrence Berkeley National Laboratory. The Venkat matrix was supplied by V. Venkatakrishnan from NASA and the Wigton matrix by L. Wigton from Boeing Commercial Airplane Group. Despite being

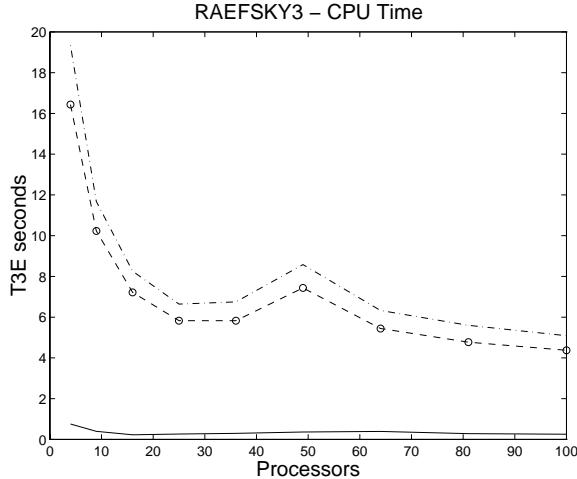


FIGURE 3. Time results on different processor numbers: total solution time (dash-dotted line), preconditioning operation time (dash-circled line), and FGMRES time (solid line).

small, the Wigton matrix is fairly difficult to solve by iterative methods. In all the tests, the right-hand sides were generated artificially by assuming that the solution is a vector of all ones.

4.1. Tests with approximate Schur-ILU preconditioning. Two test problems RAEFSKY3 and VENKAT50, described in Table 1, as well as a 5-point PDE problem are considered for the numerical experiments in this subsection. In these test problems, the matrix rows followed by the columns were scaled by 2-norm. The initial guess was set to zero. A flexible variant of restarted GMRES (FGMRES) [17] with a subspace dimension of 20 has been used to solve these problems to reduce the residual norm by 10^6 . In the preconditioning phase of the solution, the related Schur complement systems have been solved by ILUT-preconditioned GMRES, and thus preconditioning operations differed from one FGMRES iteration to another. Furthermore, varying the preconditioning parameters, such as the number of fill-in elements for ILUT, the maximum number of iterations, and tolerance for GMRES, affects the overall cost of the solution.

In general, the more accurate solves with the Schur complement system, the faster (in terms of iteration numbers) the convergence of the original system. However, even for rather large numbers of processors, preconditioning operations account for the largest amount of time spent in the iterative solution. Consider, for example, the comparison of the total solution time for RAEFSKY3 versus the preconditioning time (Figure 3). For the two test problems, Figure 4 displays the time and iteration number results with various choices (see Table 2) for the preconditioning parameters. In Table 2, each set of choices is assigned a name stated in column *Label* and the GMRES parameters tolerance and maximum number of iterations are shown in columns *tol* and *itmax*, respectively. For the Schur-ILU preconditioning, the parameter *lfil* specifies the amount of fill-ins in the whole local matrix A_i in the processor i (see equation (3)). Thus, with an increase in *lfil*, as it can be inferred from equation (13), the accuracy of the approximations to the parts of

TABLE 2. Schur-ILU preconditioning parameter choices for RAEFSKY3 and VENKAT50.

Problem	Label	<i>lfil</i>	<i>tol</i>	<i>itmax</i>
RAEFSKY3	Rprec1	40	10^{-3}	5
	Rprec2	90	10^{-4}	30
VENKAT50	Vprec1	20	10^{-3}	5
	Vprec2	50	10^{-4}	30

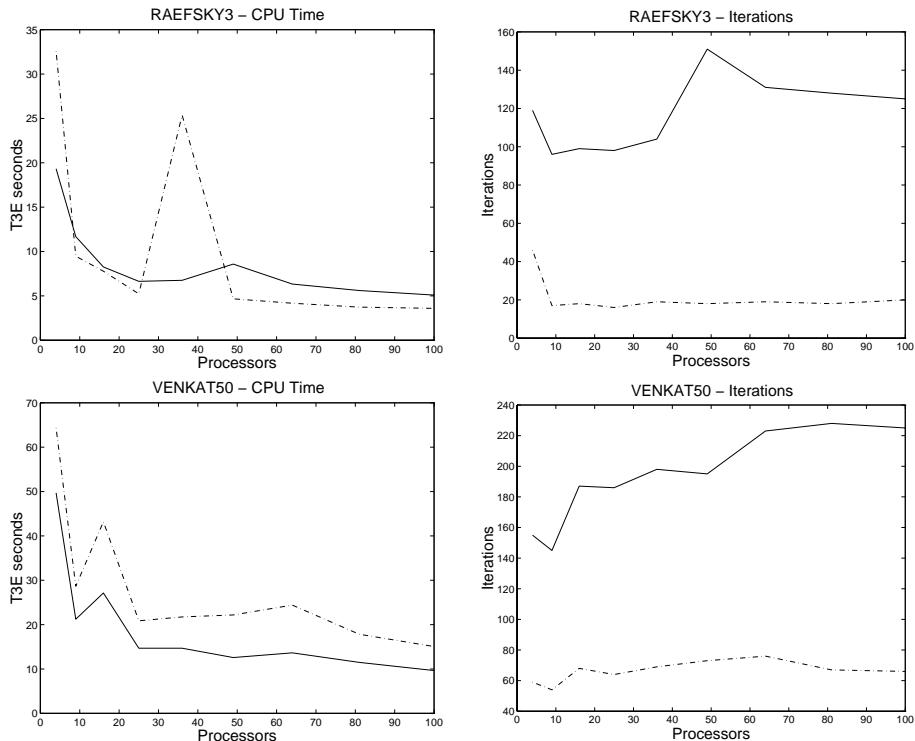


FIGURE 4. Solution time and iterations when different accuracy is used in Schur-ILU preconditioning: For RAEFSKY3, Rprec1 (solid line) and Rprec2 (dash-dotted line); for VENKAT50, Vprec1 (solid line) and Vprec2 (dash-dotted line).

L_i and U_i increases. However, an increase in the accuracy of the preconditioning operation does not necessarily lead to a smaller CPU time cost (cf., for example, the numerical results for VENKAT50 in Figure 4).

Notice the upward jump of the execution time which occurs for the RAEFSKY3 example for exactly 36 processors. This behavior is common for parallel iterative solvers for irregularly structured problems. For a reason that would be difficult to determine, the local linear systems suddenly become hard to solve for this particular number of processors, causing these solves to take the maximum number of steps (30) in each or most of the calls. The difficulty disappears as soon as we restrict the maximum number of steps in each preconditioning operation to a smaller number

(e.g., 5). Unfortunately, these difficulties are hard to predict in advance, at the time when the partitioning is performed. The problem does not occur for 37 or 35 processors.

Finally, we should point out that the previous two test matrices are relatively small for a machine of the size of the T3E, and so the fact that the execution times do not decrease substantially beyond 20 or 30 processors should not be too surprising.

Next we consider a linear system which arises from a 2-dimensional regular mesh problem. Specifically, consider the elliptic equation:

$$-\Delta u + 100 \exp(x * y) \frac{\partial u}{\partial x} + 100 \exp(-x * y) \frac{\partial u}{\partial y} - 100u = f$$

on the square $(0, 1)^2$ with Dirichlet boundary conditions. When discretized using centered difference so that there are 720 interior mesh points in each direction, the resulting linear system is of size $n = 720^2 = 518,400$. The shift term $-100u$ makes the problem indefinite.

It is interesting to observe various measures of parallel performance for this problem. Strictly speaking, each run is different since the preconditioners are different, resulting from different partitionings. In particular, the number of iterations required to converge increases substantially from 4 processors to 100 processors, as shown in Figure 5 (top-left plot). This contrasts with the earlier example and other tests seen in [19]. Recall that the problem is indefinite. The increase in the number of iterations adds to the deterioration of the achievable speed-up for larger numbers of processors. It is informative to have a sense of how much of the loss of efficiency is due to convergence deterioration versus other factors, such as communication time, load imbalance, etc. For example, if we were to factor out the loss due to increased iteration numbers, then for 60 processors over 4 processors, the speed-up would be about 13, compared with the perfect speed-up of 15. In this case, the efficiency would be about 80%. However, the increase in the number of the iterations required for convergence reduces the speed-up to about 9 and the efficiency decreases to about 55%. The Speed-up and Efficiency plots in Figure 5 show the ‘adjusted’ measures which factor out iteration increases.

4.2. Experiments with BILUM. Standard implementations of ILUM and BILUM have been described in detail in [16, 21]. We used GMRES(10) as an accelerator for both the inner and outer iterations. The outer iteration process was preconditioned by BILUM with the dual dropping strategy discussed earlier. The inner iteration process to solve the last reduced system approximately was preconditioned by ILUT [17]. Exceptions are stated explicitly. The construction and application of the BILUM preconditioner was similar to those described in [21], except that here the dual dropping strategy was applied from the first level. The initial guess was a vector of random numbers.

The numerical experiments were conducted on a Power-Challenge XL Silicon Graphics workstation equipped with 512 MB of main memory, two 190 MHZ R10000 processors, and 1 MB secondary cache. We used FORTRAN 77 in 64-bit precision.

The inner iteration was stopped when the (inner iteration) residual in the 2-norm was reduced by a factor of 10^2 or the number of iterations exceeded 10, whichever occurred first. The outer iteration was stopped when the residual in the 2-norm was reduced by a factor of 10^7 . We also set an upper bound of 200 for the outer GMRES(10) iteration. (A symbol “–” in a table indicates that convergence

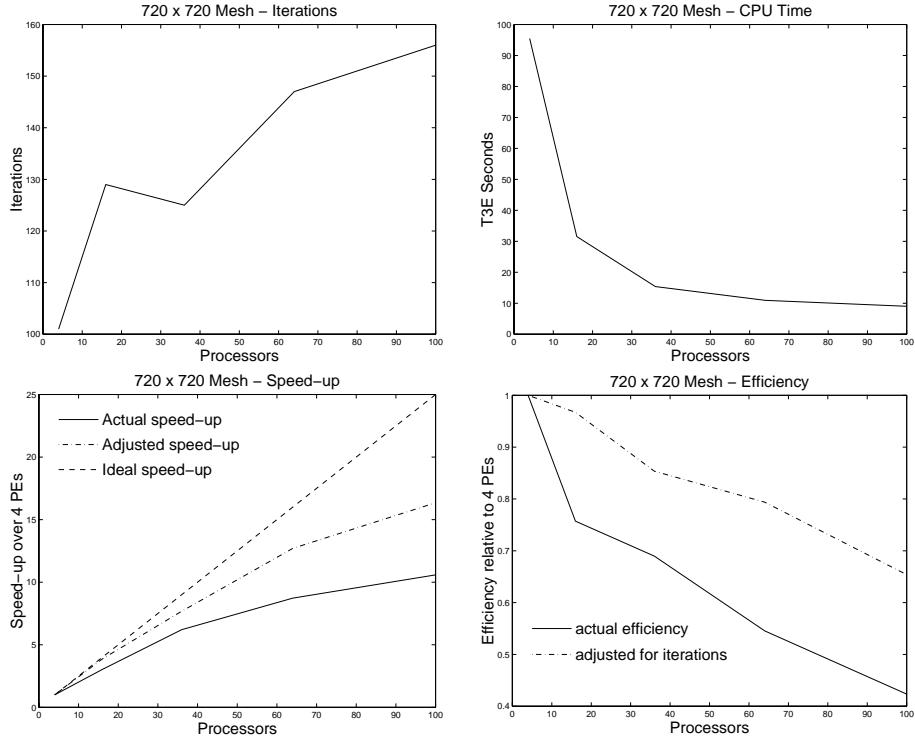


FIGURE 5. Performance measures for solving a linear system arising from a 5-point matrix on a 720×720 mesh. The adjusted speed-ups and efficiencies are defined as speed-ups and efficiencies divided by the gain (loss) ratios in the iteration count.

TABLE 3. Characteristic parameters for solving the RAEFSKY3 matrix for different sizes of uniform blocks. BILUM with 20 levels and double dropping strategy was used.

$lfil = 40, droptol = 10^{-3}$					$lfil = 50, droptol = 10^{-3}$					
k	iter.	tot-cpu	e-ratio	s-ratio	r-ratio	iter.	tot-cpu	e-ratio	s-ratio	r-ratio
5	—	—	—	1.15	0.247	—	—	—	1.33	0.258
10	—	—	—	1.44	0.213	22	90.35	2.47	1.62	0.216
15	84	140.73	0.45	1.55	0.179	17	72.71	2.70	1.72	0.178
20	150	222.60	0.28	1.83	0.156	19	81.89	2.75	2.02	0.158
25	106	160.58	0.39	1.90	0.140	16	71.19	2.96	2.07	0.129
30	—	—	—	1.87	0.122	51	103.71	0.85	2.07	0.114
35	30	74.66	1.27	1.99	0.127	14	68.48	3.31	2.24	0.120
40	27	64.85	1.25	1.92	0.100	13	60.50	3.17	2.19	0.106
45	—	—	—	1.99	0.100	160	222.78	0.25	2.24	0.106
50	—	—	—	2.07	0.106	18	66.39	2.27	2.33	0.094

did not reach in 200 outer iterations.) We used $droptol = 10^{-3}$ and tested two values for $lfil$. The block size k varied from 5 to 50. The first test was with the matrix RAEFSKY3. The test results are presented in Table 3.

TABLE 4. Characteristic parameters for solving the VENKAT50 matrix for different sizes of uniform blocks. BILUM with 10 levels and double dropping strategy was used.

$lfil = 30, droptol = 10^{-3}$					$lfil = 40, droptol = 10^{-3}$					
k	iter.	tot-cpu	e-ratio	s-ratio	r-ratio	iter.	tot-cpu	e-ratio	s-ratio	r-ratio
5	—	—	—	2.53	0.218	—	—	—	2.30	0.260
10	180	458.11	0.27	3.10	0.146	136	426.62	0.40	3.55	0.170
15	166	412.81	0.26	3.43	0.109	126	377.43	0.41	3.94	0.130
20	157	373.77	0.26	3.42	0.087	124	368.74	0.39	4.27	0.110
25	166	382.25	0.23	3.52	0.074	115	324.93	0.41	4.40	0.095
30	192	397.33	0.19	2.91	0.060	121	334.93	0.37	4.10	0.079
35	180	374.07	0.19	2.97	0.054	109	306.89	0.40	4.16	0.073
40	179	366.01	0.19	2.95	0.048	110	302.77	0.37	4.21	0.066
45	161	333.82	0.20	2.99	0.042	115	308.04	0.35	4.23	0.056
50	174	353.54	0.18	3.00	0.040	105	287.42	0.37	4.31	0.052

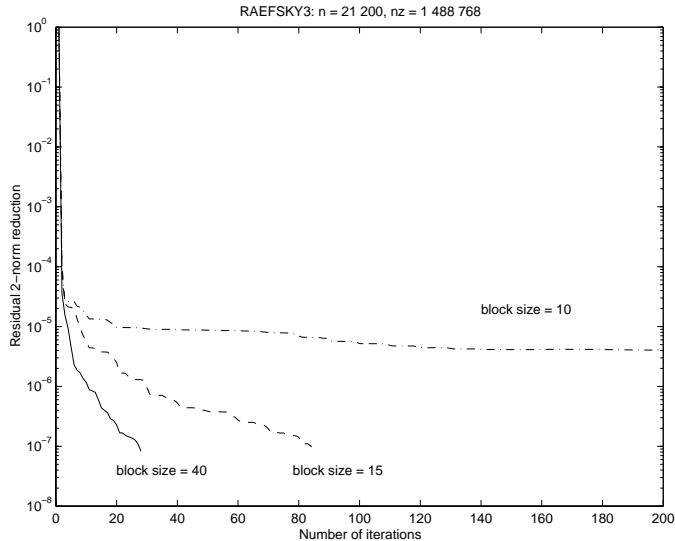


FIGURE 6. Convergence history of BILUM with different block sizes to solve the RAEFSKY3 matrix. BILUM with 20 levels and double dropping strategy was used and the coarsest level solve was preconditioned by ILUT($10^{-3}, 40$).

The results suggest that in order to increase the size of the independent set, it is preferable to have large enough blocks. However, block sizes should not be too large. In the present tests it seems that a good upper limit is the average number of nonzero elements kept in each row (the $lfil$ parameter) during the BILUM factorization. Figure 6 shows the convergence history of BILUM with different block sizes for solving the RAEFSKY3 matrix. With these block sizes, initial convergence was fast in all cases. However, after a few iterations, only BILUM using large block sizes was able to continue converging at a good rate.

TABLE 5. Test results of WIGTO966 matrix solved with ILUT.

<i>lfil</i>	<i>droptol</i>	iter.	tot-cpu	e-ratio	s-ratio
340	10^{-5}	110	81.54	2.48	9.14
350	10^{-5}	41	86.16	6.78	9.33
400	10^{-5}	22	87.07	12.60	10.06
340	10^{-4}	—	—	—	—
330	10^{-5}	—	—	—	—
330	10^{-6}	—	—	—	—

TABLE 6. Test results of WIGTO966 matrix with 10 level BILUM.

<i>k</i>	<i>lfil</i>	<i>droptol</i>	iter.	tot-cpu	e-ratio	s-ratio	r-ratio
100	50	10^{-4}	100	18.72	0.47	3.21	0.017
100	60	10^{-4}	—	—	—	—	—
100	100	10^{-4}	22	15.90	3.11	5.34	0.042
80	80	10^{-4}	—	—	—	—	—
62	50	10^{-4}	—	—	—	—	—
62	50	10^{-5}	—	—	—	—	—
62	62	10^{-4}	40	15.10	1.60	3.83	0.101
62	65	10^{-4}	42	16.50	1.59	3.91	0.101
62	70	10^{-4}	—	—	—	—	—
50	50	10^{-4}	44	12.64	1.30	3.12	0.094
50	50	10^{-3}	48	13.02	1.18	3.08	0.094
50	60	10^{-3}	35	13.48	1.85	3.50	0.081
50	60	10^{-4}	32	13.10	2.02	3.52	0.081
40	50	10^{-4}	52	14.80	1.21	3.10	0.120
40	60	10^{-4}	28	14.07	2.58	3.57	0.099
30 [†]	40	10^{-4}	69	14.71	0.91	2.65	0.006

[†]: 20 levels of reduction were used.

The second test is with the matrix VENKAT50. Each row has 28 nonzeros on average. We again used $droptol = 10^{-3}$ and tested two values for *lfil*. The test results are shown in Table 4. The test results for both RAEFSKY3 and VENKAT50 indicate that *lfil* should be chosen large enough to allow a sufficient amount of fill-ins and the block size *k* should also be large enough to insure a good reduction rate. We point out that both tests yielded slow convergence for *k* = 5 which indicates that the independent sets were not large enough to guarantee a fast convergence.

Each row of the matrix WIGTO966 has 62 nonzeros on average. This matrix is very hard to solve by ILUT [6] which only worked with a large value of *lfil*, i.e., a large number of elements per row. Test results using ILUT with different *lfil* and *droptol* are given in Table 5. We note the large values for the efficiency ratio and the sparsity ratio. Table 6 shows the test results for the matrix WIGTO966 solved by BILUM with 10 levels of reduction (an exception was indicated explicitly). Different block size *k*, *lfil*, and *droptol* were tested. We find that for this example, BILUM was 6 times faster than ILUT and used only one-third of the storage space required by ILUT. Once again, we see that the block size *k* should not be larger than the number of nonzeros kept in each row.

The option of making the diagonal blocks sparse at each level is not tested in this paper. Its potential success obviously depends on the diagonal dominance of the submatrices. There are other techniques that may be used to enhance stability of the inverse of the blocks so that preconditioning effects may be improved. A typical example is the employment of approximate singular value decomposition. Special blocks such as arrow-head matrices could also be constructed which would entail no additional fill-in in the inverse [8]. Furthermore, it is not necessary that all blocks should be of the same size. For unstructured matrices, blocks with variable sizes may be able to capture more physical information than those with uniform size. A major difficulty of applying these and other advanced techniques is the complexity of programming. We will examine these and other ideas experimentally and the results will be reported elsewhere.

5. Concluding Remarks

We discussed two techniques based on Schur complement ideas for deriving preconditioners for general sparse linear systems. The Schur-ILU preconditioning is an efficient yet simple to implement preconditioner aimed at distributed sparse linear systems. The simplicity of this preconditioner comes from the fact that only local data structures are used. The multi-level domain-type algorithm (BILUM) for solving general sparse linear systems is based on a recursive application of Schur complement techniques using small subdomains. We proposed a dual dropping strategy to improve sparsity in the BILUM factors. Several parameters were introduced to characterize the efficiency of a preconditioner. Numerical results showed that the proposed strategy works well for reducing the storage cost of BILUM with large block sizes. This class of methods offers a good alternative to the standard ILU preconditioners with threshold, in view of their robustness and efficiency. However, their implementation on parallel platforms may prove to be more challenging than the single-level Schur complement preconditioners such as the approximate Schur-LU technique.

References

1. R. E. Bank and C. Wagner, *Multilevel ILU decomposition*, Tech. report, Department of Mathematics, University of California at San Diego, La Jolla, CA, 1997.
2. T. Barth, T. F. Chan, and W.-P. Tang, *A parallel algebraic non-overlapping domain decomposition method for flow problems*, Tech. report, NASA Ames Research Center, Moffett Field, CA, 1998, In preparation.
3. E. F. F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F. W. Wubs, and P. M. de Zeeuw, *How fast the Laplace equation was solved in 1995*, Appl. Numer. Math. **32** (1997), 439–455.
4. E. F. F. Botta and F. W. Wubs, *MRILU: It's the preconditioning that counts*, Tech. Report W-9703, Department of Mathematics, University of Groningen, The Netherlands, 1997.
5. T. F. Chan, S. Go, and J. Zou, *Multilevel domain decomposition and multigrid methods for unstructured meshes: algorithms and theory*, Tech. Report 95-24, Department of Mathematics, University of California at Los Angeles, Los Angeles, CA, 1995.
6. A. Chapman, Y. Saad, and L. Wigton, *High-order ILU preconditioners for CFD problems*, Tech. Report UMSI 96/14, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 1996.
7. V. Eijkhout and T. Chan, *ParPre a parallel preconditioners package, reference manual for version 2.0.17*, Tech. Report CAM Report 97-24, Department of Mathematics, University of California at Los Angeles, Los Angeles, CA, 1997.
8. G. H. Golub and J. M. Ortega, *Scientific computing: An introduction with parallel computing*, Academic Press, Boston, 1993.

9. W. D. Gropp and B. Smith, *User's manual for KSP: data-structure neutral codes implementing Krylov space methods*, Tech. Report ANL-93/23, Argonne National Laboratory, Argonne, IL, 1993.
10. S. A. Hutchinson, J. N. Shadid, and R. S. Tuminaro, *Aztec user's guide. version 1.0*, Tech. Report SAND95-1559, Sandia National Laboratory, Albuquerque, NM, 1995.
11. M. T. Jones and P. E. Plassmann, *BlockSolve95 users manual: Scalable library software for the solution of sparse linear systems*, Tech. Report ANL-95/48, Argonne National Laboratory, Argonne, IL, 1995.
12. J. Mandel, *Balancing domain decomposition*, Comm. Appl. Numer. Methods **9** (1993), 233–241.
13. A. A. Reusken, *Approximate cyclic reduction preconditioning*, Tech. Report RANA 97-02, Department of Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands, 1997.
14. J. W. Ruge and K. Stüben, *Efficient solution of finite difference and finite element equations*, Multigrid Methods for Integral and Differential Equations (Oxford) (D. J. Paddon and H. Holstein, eds.), Clarendon Press, 1985, pp. 169–212.
15. Y. Saad, *Parallel sparse matrix library (P-SPARSLIB): The iterative solvers module*, Advances in Numerical Methods for Large Sparse Sets of Linear Equations (Yokohama, Japan), vol. Number 10, Matrix Analysis and Parallel Computing, PCG 94, Keio University, 1994, pp. 263–276.
16. ———, *ILUM: a multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput. **17** (1996), no. 4, 830–847.
17. ———, *Iterative methods for sparse linear systems*, PWS Publishing, New York, 1996.
18. Y. Saad and A. Mallevsky, *PSPARSLIB: A portable library of distributed memory sparse iterative solvers*, Proceedings of Parallel Computing Technologies (PaCT-95), 3-rd international conference (St. Petersburg, Russia) (V. E. Malyshkin et al., ed.), 1995.
19. Y. Saad and M. Sosonkina, *Distributed Schur complement techniques for general sparse linear systems*, Tech. Report UMSI 97/159, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 1997.
20. Y. Saad and K. Wu, *Design of an iterative solution module for a parallel sparse matrix library (P-SPARSLIB)*, Proceedings of IMACS Conference, 1994 (Georgia) (W. Schonauer, ed.), 1995.
21. Y. Saad and J. Zhang, *BILUM: block versions of multi-elimination and multi-level ILU preconditioner for general sparse linear systems*, Tech. Report UMSI 97/126, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 1997.
22. B. Smith, P. Bjørstad, and W. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, New York, NY, 1996.
23. B. Smith, W. D. Gropp, and L. C. McInnes, *PETSc 2.0 user's manual*, Tech. Report ANL-95/11, Argonne National Laboratory, Argonne, IL, 1995.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MN 55455

E-mail address: saad@cs.umn.edu

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF MINNESOTA – DULUTH, DULUTH, MN 55812–2496

E-mail address: masha@d.umn.edu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MN 55455

E-mail address: jzhang@cs.umn.edu

Spectral/ hp Methods For Elliptic Problems on Hybrid Grids

Spencer J. Sherwin, Timothy C.E. Warburton,
and George Em Karniadakis

1. Introduction

We review the basic algorithms of spectral/ hp element methods on tetrahedral grids and present newer developments on hybrid grids consisting of tetrahedra, hexahedra, prisms, and pyramids. A unified tensor-product trial basis is developed for all elements in terms of non-symmetric Jacobi polynomials. We present in some detail the patching procedure to ensure C^0 continuity and appropriate solution techniques including a multi-level Schur complement algorithm.

In standard low-order methods the quality of the numerical solution of an elliptic problem depends critically on the grid used, especially in three-dimensions. Moreover, the efficiency to obtain this solution depends also on the grid, not only because grid generation may be the most computationally intensive stage of the solution process but also because it may dictate the efficiency of the parallel solver to invert the corresponding algebraic system. It is desirable to employ grids which can handle arbitrary geometric complexity and exploit existing symmetry and structure of the solution and the overall domain.

Tetrahedral grids provide great flexibility in complex geometries but because of their unstructured nature they require more memory compared with structured grids consisting of hexahedra. This extra memory is used to store connectivity information as well as the larger number of tetrahedra required to fill a specific domain, i.e. five to six times more tetrahedra than hexahedra. From the parallel solver point of view, large aspect ratio tetrahedra can lead to substantial degradation of convergence rate in iterative solvers, and certain topological constraints need to be imposed to maintain a balanced parallel computation.

The methods we discuss in this paper address both of the aforementioned issues. First, we develop high-order hierarchical expansions with exponential convergence for smooth solutions, which are substantially less sensitive to grid distortions. Second, we employ hybrid grids consisting of tetrahedra, hexahedra, prisms, and pyramids that facilitate great discretisation flexibility and lead to substantial memory savings. An example of the advantage of hybrid grids was reported in [5] where only 170K tetrahedra in combination with prisms were employed to construct a hybrid grid around the high-speed-civil-transport aircraft instead of an estimated

1991 *Mathematics Subject Classification*. Primary 65N30; Secondary 65N50.

two million if tetrahedra were used everywhere instead of triangular prisms. In general, for elliptic problems with steep boundary layers *hybrid* discretisation is the best approach in accurately resolving the boundary layers while efficiently handling any geometric complexities.

In previous work [13, 14] we developed a spectral/hp element method for the numerical solution of the two- and three-dimensional unsteady Navier-Stokes equations. This formulation was implemented in the parallel code ***NekTar*** [11]. The discretisation was based on arbitrary triangulations/ tetrahedrisations of (complex-geometry) domains. On each triangle or tetrahedron a spectral expansion basis is employed consisting of Jacobi polynomials of mixed weight that accommodate exact numerical quadrature. The expansion basis is hierarchical of variable order per element and retains the tensor product property (similar to standard spectral expansions), which is key in obtaining computational efficiency via the sum factorisation technique. In addition to employing standard tetrahedral grids for discretisation, the formulation employed is also based on standard finite element concepts. For example, the expansion basis is decomposed into vertex modes, edge modes, face modes and interior modes as in other hexahedral h-p bases [16, 8]. With this decomposition, the C^0 continuity requirement for second-order elliptic problems is easily implemented following a direct stiffness assembly procedure.

In this paper, we extend the preliminary work of [10] in formulating a unified hierarchical hybrid basis for multiple domains. Specifically, we describe the basis in tensor-product form using a new coordinate system and provide details on how these heterogeneous subdomains can be patched together. We then concentrate on investigating the scaling of the condition number of the Laplacian system and discuss solution techniques, including a multi-level Schur complement algorithm [15].

2. Unified Hybrid Expansion Bases

In this section we shall develop a unified hybrid expansion basis suitable for constructing a C^0 global expansion using triangular and quadrilateral regions in two-dimensions and tetrahedral, pyramidal, prismatic and hexahedral domains in three-dimensions. This unified approach lends itself naturally to an object orientated implementation as originally developed in [17] using C++ in the code ***NekTar***. To construct these expansion we must first introduce an appropriate coordinate system as discussed in section 2.1. Having developed the coordinate system the definition of the basis in terms of Jacobi polynomials is outlined in section 2.2.

2.1. Coordinate Systems. We define the standard quadrilateral region as

$$\mathcal{Q}^2 = \{(\xi_1, \xi_2) | -1 \leq \xi_1, \xi_2 \leq 1\},$$

within which we note that the Cartesian coordinates (ξ_1, ξ_2) are bounded by constant limits. This is not, however, the case in the standard triangular region defined as

$$\mathcal{T}^2 = \{(\xi_1, \xi_2) | -1 \leq \xi_1, \xi_2; \xi_1 + \xi_2 \leq 0\}.$$

where the bounds of the Cartesian coordinates (ξ_1, ξ_2) are clearly dependent upon each other. To develop a suitable tensorial type basis within unstructured regions, such as the triangle, we need to develop a new coordinate system where the local

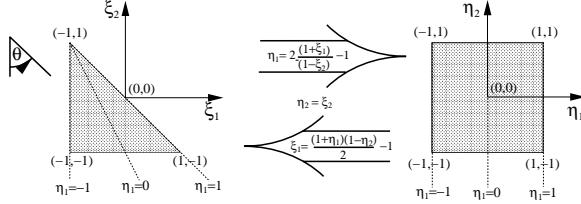


FIGURE 1. Triangle to rectangle transformation.

coordinates have independent bounds. The advantage of such a system is that we can then define one-dimensional functions upon which we can construct our multi-domain tensorial basis. It also defines an appropriate system upon which we can perform important numerical operations such as integration and differentiation [6].

2.1.1. Collapsed Two-Dimensional Coordinate System. A suitable coordinate system, which describes the triangular region between constant independent limits, is defined by the transformation

$$(1) \quad \begin{aligned} \eta_1 &= 2\frac{(1+\xi_1)}{(1-\xi_2)} - 1 \\ \eta_2 &= \xi_2, \end{aligned}$$

and has the inverse transformation

$$(2) \quad \begin{aligned} \xi_1 &= \frac{(1+\eta_1)(1-\eta_2)}{2} - 1 \\ \xi_2 &= \eta_2, \end{aligned}$$

These new local coordinates (η_1, η_2) define the standard triangular region by

$$\mathcal{T}^2 = \{(\eta_1, \eta_2) | -1 \leq \eta_1, \eta_2 \leq 1\}.$$

The definition of the triangular region in terms of the coordinate system (η_1, η_2) is identical to the definition of the standard quadrilateral region in terms of the Cartesian coordinates (ξ_1, ξ_2) . This suggests that we can interpret the transformation (1) as a mapping from the triangular region to a rectangular one as illustrated in figure 1. For this reason, we shall refer to the coordinate system (η_1, η_2) as the *collapsed coordinate system*. Although this transformation introduces a multi-values coordinate (η_1) at $(\xi_1 = -1, \xi_2 = 1)$, we note that singular point of this nature commonly occur in cylindrical and spherical coordinate systems.

2.1.2. Collapsed Three-Dimensional Coordinate Systems. The interpretation of a triangle to rectangle mapping of the two-dimensional local coordinate system, as illustrated in figure 1, is helpful in the construction of a new coordinate system for three-dimensional regions. If we consider the local coordinates (η_1, η_2) as independent axes (although they are not orthogonal), then the coordinate system spans a rectangular region. Therefore, if we start with a rectangular region, or hexahedral region in three-dimensions, and apply the inverse transformation (2) we can derive a new local coordinate system in the triangular region \mathcal{T}^2 , or tetrahedron region \mathcal{T}^3 in three-dimensions, where \mathcal{T}^3 is defined as:

$$\mathcal{T}^3 = \{-1 \leq \xi_1, \xi_2, \xi_3; \xi_1 + \xi_2 + \xi_3 \leq -1\}.$$

To reduce the hexahedron to a tetrahedron requires repeated application of the transformation in (2) as illustrated in figure 2. Initially, we consider a hexahedral

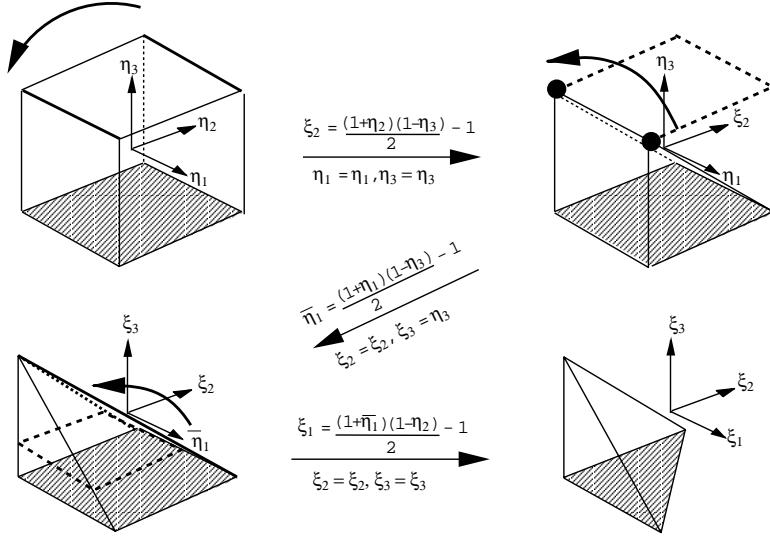


FIGURE 2. Hexahedron to tetrahedron transformation by repeatedly applying the rectangle to triangle mapping (2).

domain defined in terms of the local coordinate system (η_1, η_2, η_3) where all three coordinates are bounded by constant limits, i.e. $(-1 \leq \eta_1, \eta_2, \eta_3 \leq 1)$. Applying the rectangle to triangle transformation (2) in the (η_2, η_3) we obtain a new ordinate (ξ_2) such that

$$\xi_2 = \frac{(1 + \eta_2)(1 - \eta_3)}{2} - 1 \quad \eta_3 = \eta_3.$$

Treating the coordinates (η_1, ξ_2, η_3) as independent, the region which originally spanned a hexahedral domain is mapped to a rectangular prism. If we now apply transformation (2) in the (η_1, η_3) plane, introducing the ordinates $\bar{\eta}_1, \xi_3$ defined as

$$\bar{\eta}_1 = \frac{(1 + \eta_1)(1 - \eta_3)}{2} - 1 \quad \xi_3 = \eta_3,$$

we see that the coordinates $(\bar{\eta}_1, \xi_2, \xi_3)$ span a region of a square based pyramid. The third and final transformation to reach the tetrahedral domain is a little more complicated as to reduce the pyramidal region to a tetrahedron we need to apply the mapping in every square cross section parallel to the $(\bar{\eta}_1, \xi_2)$ plane. This means using the transformation (2) in the $(\bar{\eta}_1, \xi_2)$ plane to define the final ordinate (ξ_1) as

$$\xi_1 = \frac{(1 + \bar{\eta}_1)(1 - \xi_2)}{2} - 1 \quad \xi_2 = \xi_2.$$

If we choose to define the coordinate of the tetrahedron region (ξ_1, ξ_2, ξ_3) as the orthogonal Cartesian system then, by determining the hexahedral coordinates (η_1, η_2, η_3) in terms of the orthogonal Cartesian system, we obtain

$$(3) \quad \eta_1 = 2 \frac{(1 + \xi_1)}{(-\xi_2 - \xi_3)} - 1, \quad \eta_2 = 2 \frac{(1 + \xi_2)}{(1 - \xi_3)} - 1, \quad \eta_3 = \xi_3,$$

TABLE 1. The local Collapsed Cartesian coordinates which have constant bounds within the standard region may be expressed in terms of the Cartesian coordinates ξ_1, ξ_2, ξ_3 . Each region may be defined in terms of the local coordinates since having a lower bound of $-1 \leq \xi_1, \xi_2, \xi_3$ and upper bound as indicated in the table. Each region and the planes of constant local coordinate are shown in figure 3.

Region	Upper bound	Local Coordinate		
Hexahedron	$\xi_1, \xi_2, \xi_3 \leq 1$	ξ_1	ξ_2	ξ_3
Prism	$\xi_1, \xi_2 + \xi_3 \leq 1$	ξ_1	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	ξ_3
Pyramid	$\xi_1 + \xi_3, \xi_2 + \xi_3 \leq 1$	$\bar{\eta}_1 = \frac{2(1+\xi_1)}{(1-\xi_3)} - 1$	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	$\eta_3 = \xi_3$
Tetrahedron	$\xi_1 + \xi_2 + \xi_3 \leq 1$	$\eta_1 = \frac{2(1+\xi_1)}{(-\xi_2-\xi_3)} - 1$	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	$\eta_3 = \xi_3$

which is a new local coordinate system for the tetrahedral domain which is bounded by constant limits. When $\xi_3 = -1$ this system reduces to the two-dimensional system defined in (1).

In a similar manner, if we had chosen to define the coordinates in either the pyramidal or prismatic region as the orthogonal Cartesian system then evaluating the hexahedral coordinates in terms of these coordinates would generate a new local collapsed system for these domains. Table 1 shows the local collapsed coordinate systems in all the three-dimensional regions. A diagrammatic representation of the local collapsed coordinate system is shown in figure 3.

2.2. C^0 Continuous Unstructured Expansions. Ideally we would like to use an elemental expansion which can be assembled into a globally orthogonal expansion. Although it is possible to derive an orthogonal expansion within an elemental region [12, 10], the requirement to easily impose boundary conditions and tessellate the local expansions into multiple domains necessitates some modifications which destroy the orthogonality. To construct a C^0 continuous bases we decompose orthogonal expansion developed in [12, 10] into an interior and boundary contribution as is typical of all hp finite element methods [16]. The interior modes (or bubble functions) are defined as zero on the boundary of the local domain. The completeness of the expansion is then ensured by adding boundary modes which consist of vertex, edge and face contributions. The vertex modes have unit value at one vertex and decay to zero at all other vertices; edge modes have local support along one edge and are zero on all other edges and vertices; face modes have local support on one face and are zero on all other faces, edges and vertices. Using this decomposition C^0 continuity between elements can be enforced by matching similar shaped boundary modes providing some orientation constraints are satisfied as discussed in section 3. To construct the unified hybrid expansions we shall initially define a set of principal functions in section 2.2.1. Using these functions we then define the construction of the expansions in sections 2.2.2 and 2.2.3.

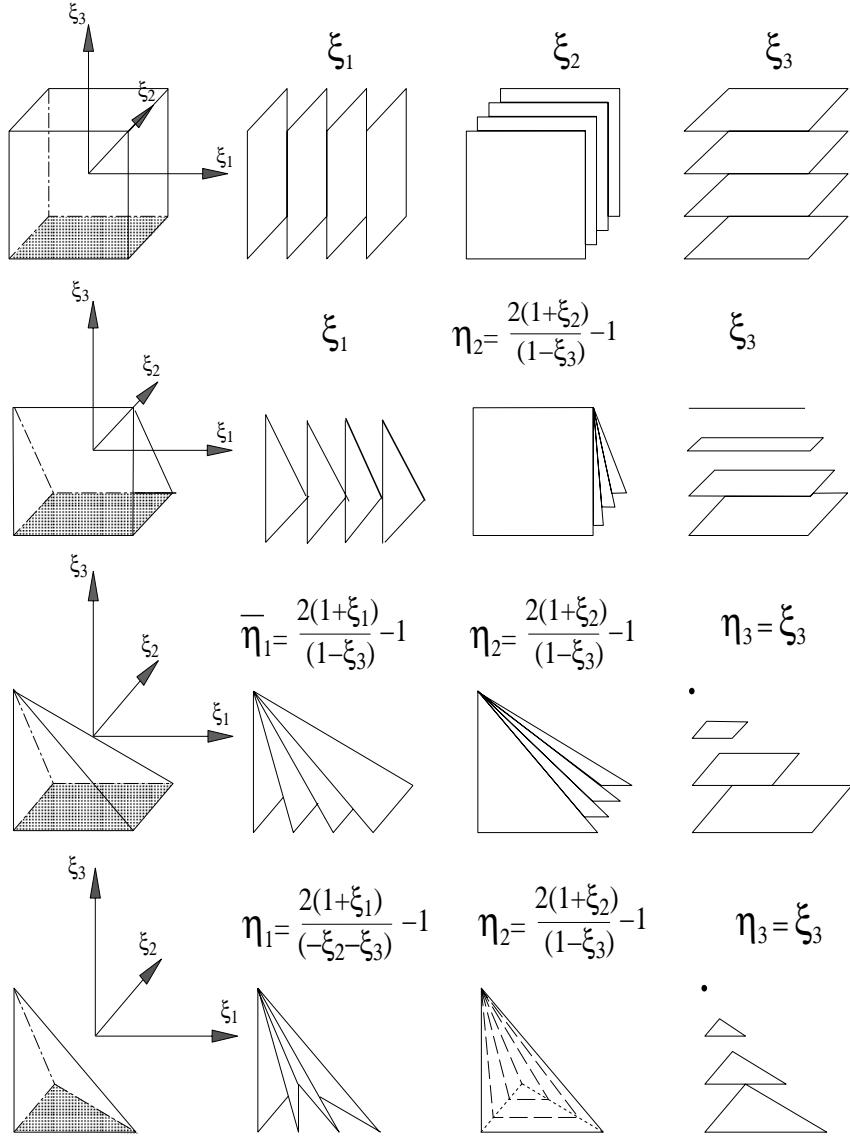


FIGURE 3. Planes of constant value of the local collapsed Cartesian coordinate systems in the hexahedral, prismatic, pyramidal and tetrahedral domains. In all but the hexahedral domain, the standard Cartesian coordinates ξ_1, ξ_2, ξ_3 describing the region have an upper bound which couples the coordinate system as shown in table 1. The local collapsed Cartesian coordinate system $\eta_1, \bar{\eta}_1, \eta_2, \eta_3$ represents a system of non-orthogonal coordinates which are bounded by a constant value within the region.

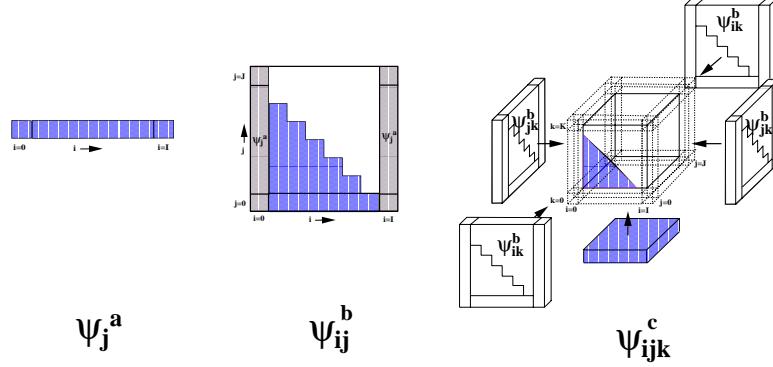


FIGURE 4. Illustration of the structure of the arrays of principal functions $\psi_i^a(z)$, $\psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$. These arrays are not globally closed packed although any edge, face or interior region of the array may be treated as such. The interior of the arrays $\psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ have been shaded to indicate the minimum functions required for a complete triangular and tetrahedral expansion.

2.2.1. *Principal Functions.* Denoting $P_i^{\alpha,\beta}(z)$ as the i^{th} order Jacobi polynomial which satisfied the orthogonality condition

$$\int_{-1}^1 (1-z)^\alpha (1+z)^\beta P_i^{\alpha,\beta}(z) P_j^{\alpha,\beta}(z) dz = C \delta_{ij} \quad \text{where } \alpha, \beta > -1,$$

we define three principal functions denoted by $\psi_i^a(z)$, $\psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ ($0 \leq i \leq I$, $0 \leq j \leq J$, $0 \leq k \leq K$) :

$$\begin{aligned} \psi_i^a(z) &= \begin{cases} \left(\frac{1-z}{2}\right)^i & i = 0 \\ \left(\frac{1-z}{2}\right)^i \left(\frac{1+z}{2}\right) P_{i-1}^{1,1}(z) & 1 \leq i \leq I-1 \\ \left(\frac{1+z}{2}\right)^I & i = I \end{cases}, \\ \psi_{ij}^b(z) &= \begin{cases} \psi_j^a(z) & i = 0, \quad 0 \leq j \leq J \\ \left(\frac{1-z}{2}\right)^{i+1} & 1 \leq i \leq I-1, \quad j = 0 \\ \left(\frac{1-z}{2}\right)^{i+1} \left(\frac{1+z}{2}\right) P_{j-1}^{2i+1,1}(z) & 1 \leq i \leq I-1, \quad 1 \leq j \leq J-1 \\ \psi_j^a(z) & i = I, \quad 0 \leq j \leq J \end{cases}, \\ \psi_{ijk}^c(z) &= \begin{cases} \psi_{jk}^b(z) & i = 0, \quad 0 \leq j \leq J, \quad 0 \leq k \leq K \\ \psi_{ik}^b(z) & 0 \leq i \leq I, \quad j = 0, \quad 0 \leq k \leq K \\ \left(\frac{1-z}{2}\right)^{i+j+1} & 1 \leq i \leq I-1, \quad 1 \leq j \leq J-1, \quad k = 0 \\ \left(\frac{1-z}{2}\right)^{i+j+1} \left(\frac{1+z}{2}\right) P_{k-1}^{2i+2j+1,1}(z) & 1 \leq i \leq I-1, \quad 1 \leq j \leq J-1, \quad 1 \leq k \leq K-1 \\ \psi_{ik}^b(z) & 0 \leq i \leq I, \quad j = J, \quad 0 \leq k \leq K \\ \psi_{jk}^b(z) & i = I, \quad 0 \leq j \leq J, \quad 0 \leq k \leq K \end{cases}. \end{aligned}$$

Figure 4 diagrammatically indicates the structure of the principle functions $\psi_i^a(z)$, $\psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ as well as how the function $\psi_i^a(z)$ is incorporated into $\psi_{ij}^b(z)$, and similarly how $\psi_{ij}^b(z)$ is incorporated into $\psi_{ijk}^c(z)$. The function $\psi_i^a(z)$ has been decomposed into two linearly varying components and a function which is zero at the end points. The linearly varying components generate the vertex modes which are identical to the standard linear finite element expansion. The interior contributions of all the base functions (i.e. $1 \leq i \leq I-1$, $1 \leq j \leq J-1$, $1 \leq k \leq K-1$) are similar in form to the orthogonal basis functions defined in [10]. However, they

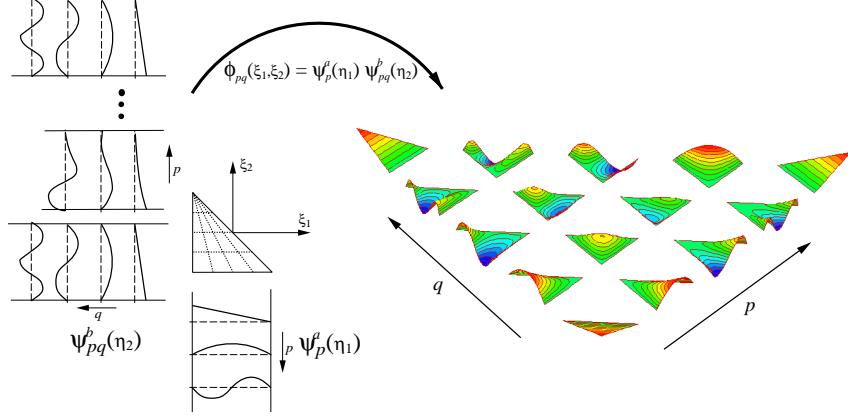


FIGURE 5. Construction of a fourth-order ($P = 4$) triangular expansion using the product of two principal functions $\psi_p^a(\eta_1)$ and $\psi_{pq}^b(\eta_2)$.

are now pre-multiplied by a factor of the form $(\frac{1-z}{2}) (\frac{1+z}{2})$ which ensures that these modes are zero on the boundaries of the domain. The value of α, β in the Jacobi polynomial $P_p^{\alpha, \beta}(x)$ has also been slightly modified to maintain as much orthogonality as possible in the mass and Laplacian systems.

2.2.2. Hybrid Expansions. The two-dimensional expansions are defined in terms of the principal functions as:

$$\text{Quadrilateral expansion: } \phi_{pq}(\xi_1, \xi_2) = \psi_p^a(\xi_1) \psi_q^a(\xi_2)$$

$$\text{Triangular expansion: } \phi_{pq}(\xi_1, \xi_2) = \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2)$$

where

$$\eta_1 = \frac{2(1 + \xi_1)}{(1 - \xi_2)} - 1, \quad \eta_2 = \xi_2,$$

are the two-dimensional collapsed coordinates. In figure 5 we see all of the modified expansion modes for a fourth-order ($P = 4$) modified triangular expansion. From this figure it is immediately evident that the interior modes have zero support on the boundary of the element. This figure also illustrates that the shape of every boundary mode along a single edge is identical to one of the modes along the other two edges and which allows the modal shapes in two regions to be globally assembled into a C^0 continuous expansion. In the three-dimensional expansion an equivalent condition is ensured by the introduction of $\psi_{ij}^b(z)$ into $\psi_{ijk}^c(z)$.

The three-dimensional expansions are defined in terms of the principal functions as:

$$\text{Hexahedral expansion: } \phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\xi_1) \psi_q^a(\xi_2) \psi_r^a(\xi_3)$$

$$\text{Prismatic expansion: } \phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\xi_1) \psi_q^a(\eta_2) \psi_{qr}^b(\xi_3)$$

$$\text{Pyramidalic expansion: } \phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\bar{\eta}_1) \psi_q^a(\eta_2) \psi_{pqr}^c(\eta_3)$$

$$\text{Tetrahedral expansion: } \phi_{pqr}(\xi_1, \xi_2, \xi_3) = \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2) \psi_{pqr}^c(\eta_3)$$

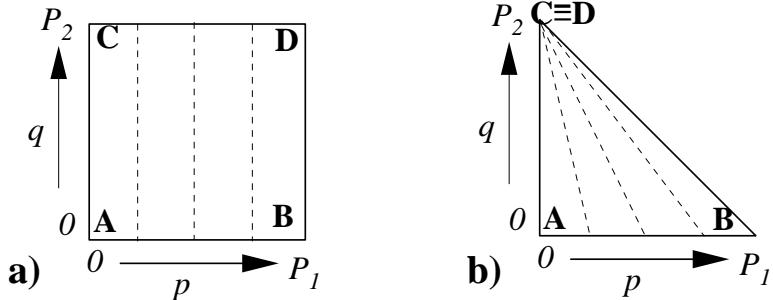


FIGURE 6. The construction of the collapsed Cartesian coordinates system maps vertex D onto vertex C in plot (a). If we consider the quadrilateral region in plot (a) as describing a two-dimensional array in p and q then we can imagine an equivalent array within the triangular region as shown in plot (b).

where

$$\eta_1 = \frac{2(1 + \xi_1)}{(-\xi_2 - \xi_3)} - 1, \quad \bar{\eta}_1 = \frac{2(1 + \xi_1)}{(1 - \xi_3)} - 1, \quad \eta_2 = \frac{2(1 + \xi_2)}{(1 - \xi_3)} - 1, \quad \eta_3 = \xi_3,$$

are the three-dimensional collapsed coordinates.

2.2.3. Construction of Basis From Principal Functions. As can be appreciated from figure 4 the principal functions for the unstructured regions are not in a closed packed form and so we cannot consecutively loop over the indices p, q and r to arrive at a complete polynomial expansion. Even though these arrays are not closed packed their definition permits an intuitive construction of the expansion basis as discussed below.

Two-Dimensions

The quadrilateral expansion may be constructed by considering the definition of the basis $\phi_{pq}(\xi_1, \xi_2)$ as a two-dimensional array within the standard quadrilateral region with the indices $p = 0, q = 0$ corresponding to the lower left hand corner as indicated in figure 6(a). Using this diagrammatic form of the array it was easy to construct the vertex and edge modes by determining the indices corresponding to the vertex or edge of interest. A similar approach is possible with the modified triangular expansion.

We recall that to construct the local coordinate system we used a collapsed Cartesian system where vertex D in figure 6(a) was collapsed onto vertex C as shown in figure 6(b). Therefore, if we use the equivalent array system in the triangular region we can construct our triangular expansions. For example, the vertices marked A and B in figure 6(b) are defined as

$$\begin{aligned} \text{Vertex A} &= \phi_{00}(\eta_1, \eta_2) &= \psi_0^a(\eta_1)\psi_{00}^b(\eta_2) \\ \text{Vertex B} &= \phi_{P_1 0}(\eta_1, \eta_2) &= \psi_{P_1}^a(\eta_1)\psi_{P_1 0}^b(\eta_2). \end{aligned}$$

The vertex at the position marked CD in figure 6(b) was formed by collapsing the vertex D onto vertex C in figure 6(a). Therefore this mode is generated by adding the contribution from the indices corresponding to the vertices C and D, i.e.

$$\text{Vertex CD} = \phi_{0P_2}(\eta_1, \eta_2) + \phi_{P_1 P_2}(\eta_1, \eta_2) = \psi_0^a(\eta_1)\psi_{0P_2}^b(\eta_2) + \psi_{P_1}^a(\eta_1)\psi_{P_1 P_2}^b(\eta_2).$$

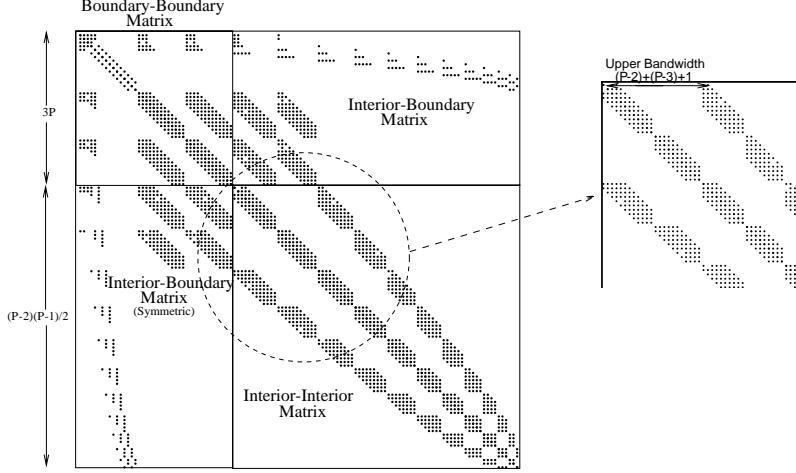


FIGURE 7. The structure of the mass matrix for a triangular expansion $\phi^{pq} = \psi_p^a \psi_{pq}^b$ of order $P_1 = P_2 = 14$ within the standard region T^2 . The boundary modes have been ordered first followed by the interior modes. If the q index is allowed to run faster, the interior matrix has a bandwidth of $(P - 2) + (P - 3) + 1$.

For the triangular expansion the edge modes are similarly defined as:

$$\begin{aligned} \text{Edge AB : } \phi_{p0}(\eta_1, \eta_2) &= \psi_p^a(\eta_1) \psi_{p0}^b(\eta_2) & (0 < p < P_1) \\ \text{Edge AC : } \phi_{0q}(\eta_1, \eta_2) &= \psi_0^a(\eta_1) \psi_{0q}^b(\eta_2) & (0 < q < P_2) \\ \text{Edge BD : } \phi_{P_1 q}(\eta_1, \eta_2) &= \psi_{P_1}^a(\eta_1) \psi_{P_1 q}^b(\eta_2) & (0 < q < P_2). \end{aligned}$$

In constructing the triangular region from the quadrilateral region as shown in figure 6 edge CD was eliminated and, as one might expect, it does not contribute to the triangular expansion.

Finally the interior modes of the modified triangular expansion (which become the triangular face modes in the three-dimensional expansions) are defined as

$$\text{Interior : } \phi^{pq}(\eta_1, \eta_2) = \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2) \quad (0 < p, q; p < P_1; p + q < P_2; P_1 \leq P_2).$$

There is a dependence of the interior modes in the p -direction on the modes in the q -direction which ensures that each mode is a polynomial in terms of the Cartesian coordinates (ξ_1, ξ_2) . This dependence requires that there should be as many modes in the q direction as there are in the p direction and hence the restriction that $P_1 \leq P_2$. A complete polynomial expansion typically involves all the modes defined above and this expansion is optimal in the sense that it spans the widest possible polynomial space in (ξ_1, ξ_2) with the minimum number of modes. More interior or edge modes could be used but if they are not increased in a consistent manner the polynomial space will not be increased. In figure 7 we see the structure of the mass matrix for a $P_1 = P_2 = 14$ polynomial order triangular expansion within the standard triangular region. The matrix is ordered so the boundary modes are

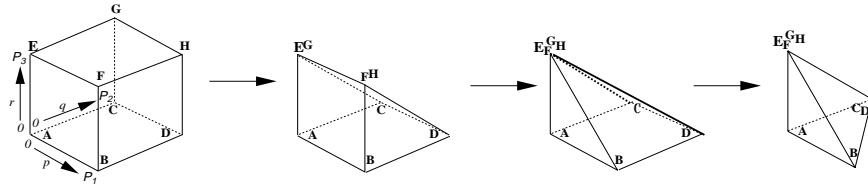


FIGURE 8. Generation of the standard tetrahedral domains from repeated collapsing of a hexahedral region.

first followed by the interior system. It can be shown (see [13]) that if we order the interior system so the q index runs fastest then the bandwidth of the interior system is $(P - 2) + (P - 3) + 1$.

Three-Dimensions

As illustrated in figure 8, for the hexahedral domain the indices p, q, r correspond directly to a three-dimensional array where all indices start from zero at the bottom left-hand corner. Therefore, the vertex mode labelled A is described by $\phi_{(000)} = \psi_0^a(\xi_1)\psi_0^a(\xi_2)\psi_0^a(\xi_3)$, similarly the vertex mode labelled H is described by $\phi_{(P_1, P_2, P_3)}$ and the edge modes between C and G correspond to $\phi_{0, P_2, r}$ ($1 < r < P_3$).

When considering the prismatic domain we use the *equivalent* hexahedral indices. Accordingly, vertex A is now described by $\phi_{(000)} = \psi_0^a(\xi_1)\psi_0^a(\eta_2)\psi_{00}^b(\xi_3)$. In generating the new coordinate system, vertex G was mapped to vertex E and therefore the vertex mode, labelled EG in the prismatic domain, is described by $\phi_{(0,0,P_3)} + \phi_{(0,P_2,P_3)}$ (i.e., adding the two vertices from the hexahedral domain which form the new vertex in the prismatic domain). A similar addition process is necessary for the prismatic edge $EG - FH$ which is constructed by adding the edge modes EF (i.e. $\phi_{(p,0,P_3)}$) to the edge modes GH (i.e., $\phi_{(p,P_2,P_3)}$). In degenerating from the hexahedral domain to the prismatic region the edges EG and FH are removed and therefore do not contribute to the prismatic expansion.

This process can also be extended to construct the expansion for the pyramidal and tetrahedral domains. For both these cases the top vertex is constructed by summing the contribution of E, F, G and H . In the tetrahedral domain edges CG and DH are also added. Although the modified functions ψ_{ij}^b and ψ_{ijk}^c are not closed packed, every individual edge, face and the interior modes may be summed consecutively.

As a final point we note that the use of the collapsed Cartesian coordinate system means that the triangular faces, unlike the quadrilateral faces, are not rotationally symmetric. This means that there is a restriction on how two triangular faces, in a multi-domain expansion, must be aligned. In section 3 we show that this condition can easily be satisfied for all tetrahedral meshes although some care must be taken when using a mixture of different elemental domains.

3. Global Assembly

The elements we have described will be tessellated together to construct a continuous solution domain. We shall only permit elements to connect by sharing common vertices, complete edges and/or complete faces. such a connectivity is

commonly referred to as conforming elements. In this section we shall discuss issues related to the process of globally assembling the elemental bases described in section 2.2.

When two elements share an edge it is important for them to be able to determine if their local coordinate system at that edge are aligned in the same direction. This information is important since it is necessary to ensure that the shape of all edges modes is similar along an edge. If the local coordinate systems are not aligned in the same direction then edges modes of odd polynomial order will have different signs and so one edge mode will need to be multiplied by -1 .

This condition becomes more complicated in three dimensions when two elements share a face. In this case it is not automatic that their coordinate systems on the common face will line up. Considering the tetrahedra we see that there is a vertex on each face that the coordinate system for that face radiates from. Similarly for the triangular faces of the prism and pyramid. We will call this vertex the face origin as it is similar to a polar coordinate origin. The alignment constraint necessitates that when two triangular faces meet their origin vertices must coincide. Initially it is not obvious how to satisfy this constraint for a mesh consisting of just tetrahedral elements. We outline two algorithms that will satisfy this constraint. The first is based on the topology of the mesh. We will only use the connections between elements to determine how we should orientate elements. In the second method we will assume that each unique vertex in the mesh will have been given a number. This second method works under some loose conditions but is extremely easy to implement and is very local in its nature.

It is useful to observe that one of the vertices of a tetrahedron is the face origin vertex for the three faces sharing that vertex. We will call this the *local top vertex*. Then there is one more face origin vertex on the remaining face which we call the *local base vertex*.

3.1. Algorithm 1. Given a conforming discretisation we can generate the local orientation of the tetrahedra using the following algorithm. We assume that we have a list of vertices and we know a list of elements which touch each vertex. This list of elements will be called a vertex group and all elements are assumed to have a tag of zero.

For every vertex in the list:

- Orientate all elements with a tag of one in this vertex group so that their *local base vertex* points at this vertex. Then set their tags to two.
- Orientate all elements with a tag of zero in this vertex group so that their *local top vertex* points at this vertex. Then set their tags to one.

This algorithm visits all vertices in the mesh and if this is the first time the elements in the vertex group have been visited the *local top vertex* is orientated at this vertex. If this is the second time the elements in the vertex group have been visited then set the *local base vertex* to this vertex. To see how this works we can consider the example shown in figure 9.

Here we assume that we are given a discretisation of a box using six tetrahedra as shown in figure 9a. Starting our algorithm we begin with vertex A. Since all elements have a tag of zero at this point we go straight to the second part of the algorithm and orientate all elements that touch this vertex so that their *local top vertices* point to A. Therefore tetrahedra HBDA and BHEA are orientated as shown in figure 9b and now have a tag set to one. Continuing to the next vertex B we

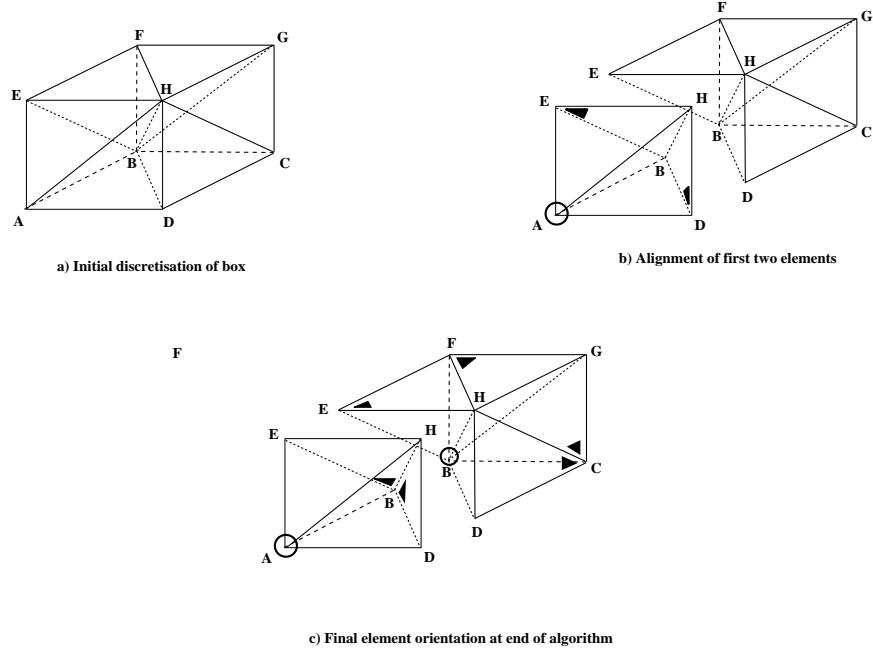


FIGURE 9. Setting up the required connectivity for the discretisation of a box as shown in a). Vertex A is given as the first *local top vertex* as shown in b). In c) vertex B is then given as the *local base vertex* and the *local base vertices* from group one are aligned to satisfy connectivity. The final element orientation is shown in figure d).

see that all elements belong to this vertex group. The first part of the algorithm is to orientate the elements with a tag of one to have their *local base vertex* pointing at B. So the tetrahedra HBDA and BHEA are rotated as shown in figure 9c and their tags are set to two. The second part of the algorithm then orientates all the other tetrahedra to have their *local top vertex* pointing at B. The connectivity is actually satisfied at this point since the orientation the faces have on the boundaries is irrelevant. However, if we continue the algorithm looping through the vertices consecutively we end up with the tetrahedra orientated as shown in figure 9d.

Clearly, the connectivity is not unique since any elements that have their *local top vertex* pointing at E can be rotated about E. However, we have demonstrated that it is possible to satisfy the connectivity requirements imposed by the co-ordinate system and thereby imply that the requirement is non-restrictive.

3.2. Algorithm 2. Assuming that every global vertex has a unique number, then for every element we have four vertices with unique global numbers:

- Place the *local top vertex* at the global vertex with the lowest global number.
- Place the *local base vertex* at the global vertex with the second lowest global number
- Orientate the last two vertices to be consistent with the local rotation of the element (typically anti-clockwise).

It has been stated before that since the coordinate systems on the faces of the tetrahedra are not symmetric that it is too difficult to use these coordinate systems. We have shown that it is possible in linear or even constant time to satisfy this constraint for any given tetrahedral mesh. This algorithm is local to each element and should be implemented at a pre-processing stage.

We now extend this approach to include meshes consisting of tetrahedra, prisms, and hexahedra. Unfortunately, in this case we find counter-examples where it is not possible to satisfy the origin alignment constraint. We have isolated the problematic cases and they are unlikely to come up when using a mesh generator.

First we deal with the case when a quadrilateral face is shared by two elements. In this instance it is sufficient to simply make the coordinate directions agree by simply reversing either face coordinate if necessary.

We now investigate over-constrained meshes. These cases can occur when prisms and tetrahedra are used together in a mesh. We will use these examples to motivate the actual algorithm we propose. The cost of this algorithm also depends linearly on the number of elements in the mesh.

It is instructive to construct a chain of prisms. This is simply a long prism, with equilateral triangular faces, divided at intervals along its length into a set of prisms connected at their triangle faces. The connectivity constraint requires that the coordinate origins of the triangular faces must meet at every prism-prism interface. This condition enforces that the collapsed edge must run in a continuous line through the edges of the prism. Now we twist the chain around in a loop and connect its triangle ends. The chain now forms a closed loop of prisms. The orientation of the end faces of the original chain must also satisfy the connectivity constraint when they meet. But we are free to choose the orientation of the faces relative to each other. However, we can make the chain into a Möbius band by twisting it around the axis along its length. In this case the connectivity cannot be satisfied without changing the mesh.

We can construct a second counter-example, this time involving one tetrahedron and two chains of prisms. We construct two chains of prisms as outlined above and we join the tetrahedra into the prism chain by connecting two of its faces to the prism chain triangular end faces. We repeat this operation again connecting the remaining two faces of the tetrahedron to the end faces of the second chain of prisms. We can now repeat the twisting of the prism loops. This over constrain the tetrahedron so that it cannot be oriented to satisfy the connectivity condition.

These two cases indicate that we cannot allow prism chains to reconnect into closed loops and still satisfy our constraints. Also we should not allow a prism chain to connect to a tetrahedron with more than one of its triangular faces. If we only consider meshes that satisfy these two constraints, then the following algorithm will satisfy the connectivity constraints:

3.3. Algorithm For Connecting Prisms And Tetrahedra.

- Find all prism chains in the mesh.
- Create a virtual connection between the faces of the tetrahedra that meet the triangular faces at each end of the chain.
- Proceed with Algorithm 2 to connect the tetrahedral mesh treating the virtual links as real connections.

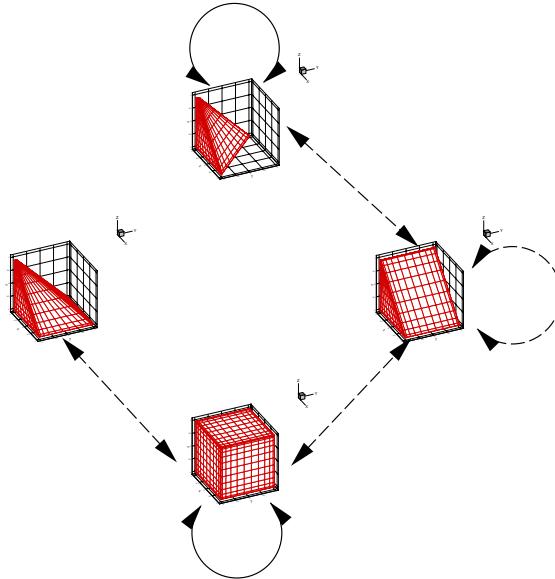


FIGURE 10. Connectivity summary. Solid arrows imply we can connect elements together with no problems, dashed arrows mean that there are some constraints on allowable configurations.

- Orient the prisms in the chains with the same orientation as the triangular faces of the tetrahedra at the ends of the virtual link. This orientation propagate through the chain.

In figure 10 we summarise which elements can be connected using the above algorithms. A solid line between elements means we can connect a given mesh with of the two element types with no problems. If the line is dashed then the mesh has to be changed to meet the connectivity constraints.

4. Global Matrix Properties

As discussed in section 2 and 3 we can construct a global expansions using a tessellated of hybrid domains which are C^0 continuous. These expansions are therefore suitable to solve second order partial differential equations using a Galerkin formulation. Consider the Helmholtz problem

$$\nabla^2 u - \lambda u = f$$

supplemented with appropriate boundary conditions. The Galerkin problem may be stated as find $u^\delta \in \mathcal{X}^\delta$ such that

$$(4) \quad a(v^\delta, u^\delta) = f(v^\delta) \quad \forall v^\delta \in \mathcal{V}^\delta,$$

where

$$\begin{aligned} a(v, u) &= \int_{\Omega} \nabla v \cdot \nabla u + \lambda v u \, d\Omega, \\ f(v) &= \int_{\Omega} v \, f \, d\Omega. \end{aligned}$$

and u^δ is the finite dimensional representation of the solution (i.e. $u^\delta = \sum \hat{u}_{pqr} \phi_{pqr}$) and $\mathcal{X}^\delta, \mathcal{V}^\delta$ are the finite dimensional space of trial and test functions. In the Galerkin approximations we assume that \mathcal{X}^δ and \mathcal{V}^δ span the same space.

As is typically of most numerical approaches, equation (4) may be represented as an algebraic system. Although these algebraic systems are typically sparse the number of degrees of freedom of a practical three-dimensional problem requires that we use an iterative solver. We are therefore interested in the conditioning of these algebraic systems. However, before considering the conditioning of this system we shall first review the restructuring of the global matrix using the static condensation technique to take advantage of the matrix structure when using spectral/hp expansions.

4.1. Matrix Solution via Schur Complement. Let us denote the global matrix problem due to the Galerkin problem (4) as

$$(5) \quad \mathbf{M}\mathbf{x} = \mathbf{f}.$$

where \mathbf{x} is a vector of global unknowns. The matrix \mathbf{M} is typically very sparse although it may have a full bandwidth. We shall assume that the global system \mathbf{M} is ordered so that the global boundary degrees of freedom are listed first, followed by the global interior degrees of freedom. In addition, we also assume that the global interior degrees of freedom were numbered consecutively. Adopting this ordering, the global matrix problem (5) can be written as

$$(6) \quad \begin{bmatrix} \mathbf{M}_b & \mathbf{M}_c \\ \mathbf{M}_c^T & \mathbf{M}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix}.$$

where we have distinguished between the boundary and interior components of \mathbf{x} and \mathbf{f} using $\mathbf{x}_b, \mathbf{x}_i$ and $\mathbf{f}_b, \mathbf{f}_i$, respectively.

The matrix \mathbf{M}_b corresponds to the global assembly of the elemental boundary-boundary mode contributions and similarly $\mathbf{M}_c, \mathbf{M}_i$ correspond to the global assembly of the elemental boundary-interior coupling and interior-interior systems. A notable feature of the global system is that the global boundary-boundary, \mathbf{M}_b , matrix is sparse and may be re-ordered to reduce the bandwidth or re-factored in a multi-level Schur Complement solver as discussed below. The global boundary-interior coupling matrix, \mathbf{M}_c , is very sparse and as we shall see may be stored in terms of its local elemental contributions. Finally, the natural form of \mathbf{M}_i is a block diagonal matrix which is very inexpensive to evaluate since each block may be inverted individually.

To solve the system (6) we can statically condense out the interior degrees of freedom by performing a block elimination. Pre-multiplying system (6) by the matrix

$$\begin{bmatrix} \mathbf{I} & -\mathbf{M}_c \mathbf{M}_i^{-1} \\ 0 & \mathbf{I} \end{bmatrix},$$

we arrive at:

$$(7) \quad \begin{bmatrix} \mathbf{M}_b - \mathbf{M}_c \mathbf{M}_i^{-1} \mathbf{M}_c^T & 0 \\ \mathbf{M}_c^T & \mathbf{M}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b - \mathbf{M}_c \mathbf{M}_i^{-1} \mathbf{f}_i \\ \mathbf{f}_i \end{bmatrix}.$$

The equation for the boundary unknowns is therefore:

$$(8) \quad (\mathbf{M}_b - \mathbf{M}_c \mathbf{M}_i^{-1} \mathbf{M}_c^T) \mathbf{x}_b = \mathbf{f}_b - \mathbf{M}_c \mathbf{M}_i^{-1} \mathbf{f}_i.$$

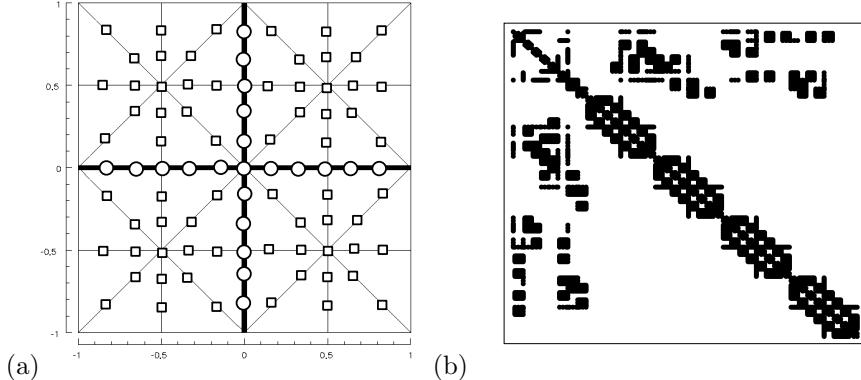


FIGURE 11. The boundary degrees of freedom, on the mesh shown in plot (a), are ordered so that the boundary modes indicated by the square symbols are first, followed by the boundary modes indicated by circular symbols within each quadrant. Using this ordering the resulting Schur complement matrix has a block diagonal sub-matrix as shown in figure (b).

Once \mathbf{x}_b is known, we can determine \mathbf{x}_i from the second row of equation (7) since

$$(9) \quad \mathbf{x}_i = \mathbf{M}_i^{-1} \mathbf{f}_i - \mathbf{M}_i^{-1} \mathbf{M}_c^T \mathbf{x}_b.$$

4.1.1. Multi-Level Schur Complement. The motivation behind using Schur complement was the natural decoupling of the interior degrees of freedom within each element leading to a global system which contained a block diagonal sub-matrix. This decoupling can be mathematically attributed to the fact that the interior degrees of freedom in one element are orthogonal to the interior degrees of freedom of another simply because these modes are non-overlapping. To take advantage of this block diagonal sub-matrix we have to construct the Schur complement system

$$\mathbf{M}_S = \mathbf{M}_b - \mathbf{M}_c [\mathbf{M}_i]^{-1} (\mathbf{M}_c)^T.$$

The effect of constructing each of this system is to orthogonalise the boundary modes from the interior modes. However, the inverse matrix $[\mathbf{M}_i]^{-1}$ is typically full, which means that the boundary modes, within an element, become tightly coupled. It is this coupling which dictates the bandwidth of the globally assembled Schur complement system. Nevertheless, an appropriate numbering of the boundary system will lead to a Schur complement matrix which also contains a sub-matrix that is block diagonal and so the static condensation technique can be re-applied. This technique has been more commonly used in the structural mechanics field and is also known as sub-structuring [15].

To illustrate this ordering we consider the triangular mesh shown in figure 11(a) using $N_{el} = 32$ elements. The construction of the global Schur complement \mathbf{M}_S requires us to globally number all of the boundary degrees of freedom as indicated by the open circles and squares. If we order the numbering of the elemental boundary degrees of freedom so that the vertex and edge modes indicated by the open circles

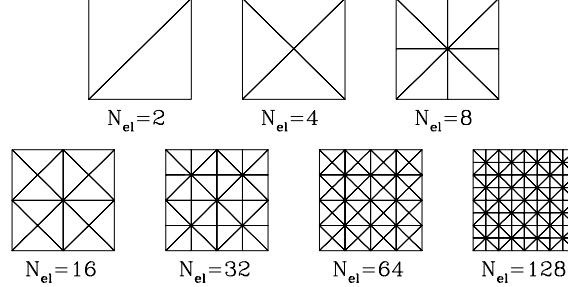


FIGURE 12. Triangulations used in determining the eigen-spectrum of the Laplacian operator.

are first followed by the vertex and edge modes within each quadrant, indicated by the open squares, then the resulting Schur complement system of the mass matrix for a polynomial expansion of $p = 6$ is shown in figure 11(b). The block diagonal structure of the matrix is due to the fact that even after constructing the elemental Schur complement systems the boundary degrees of freedom in each quadrant do not overlap and so are orthogonal.

We can now construct another Schur complement system to solve for the circle degrees of freedom and decoupling each quadrant of square degrees of freedom. This technique can be repeated providing that there is more than one region of non-overlapping data. The approach is clearly independent of the elemental shape and may equally well be applied to quadrilateral regions or any hybrid shape in three-dimensions.

4.1.2. Preconditioners. As mentioned previously, the resolution requirements for problems of practical interest typically require the use of iterative algorithms. The convergence of such algorithms depends on the eigen-spectrum of these matrices as well as the preconditioners used for convergence acceleration.

In the conjugate gradient method we estimate that the number of iterations, N_{iter} , to invert a matrix \mathbf{M} scales with the square root of the condition number, i.e.

$$N_{iter} \propto [\kappa_2(\mathbf{M})]^{1/2}.$$

The L^2 condition number of a matrix \mathbf{M} is defined as $\kappa_2(\mathbf{M}) = ||\mathbf{M}||_2 ||\mathbf{M}^{-1}||_2$, which for a symmetric matrix is equivalent to the ratio of the largest to the smallest eigenvalue.

In two-dimensions, for the *full* discrete Laplacian \mathbf{L} the condition number scales as

$$\kappa_2(\mathbf{M}) \propto N_{el} P^3.$$

The required number of iterations can therefore be very high for large problems, and especially high-order P . However, using static condensation we can consider the reduced problem consisting of the element boundary contributions by forming only the Schur complement \mathbf{M}_S . For a symmetric positive definite system the condition number of the Schur complement matrix \mathbf{M}_S can be no larger than the condition number of the complete system \mathbf{M} [15].

Experimental results indicating the scaling of $\kappa_2(\mathbf{M}_S)$ have been obtained in the domains shown in figure 12. The relationship of the condition number with

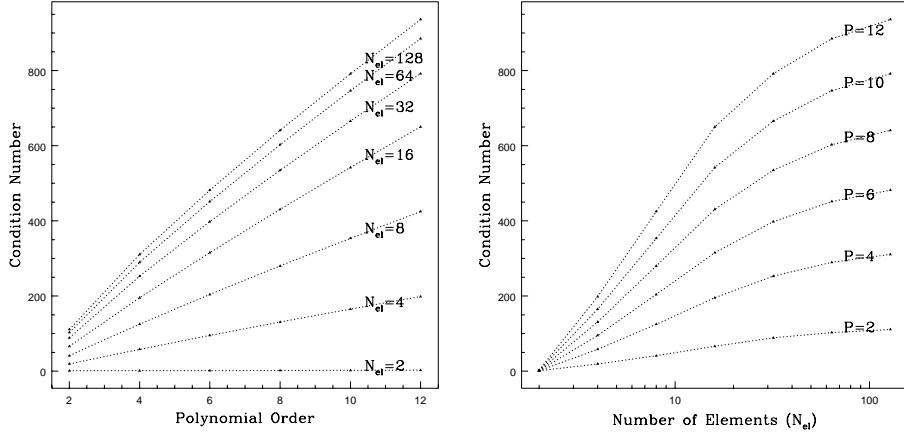


FIGURE 13. Condition number variation of the Schur complement of the discrete Laplacian operator with respect to the order (left) and the number of elements (right).

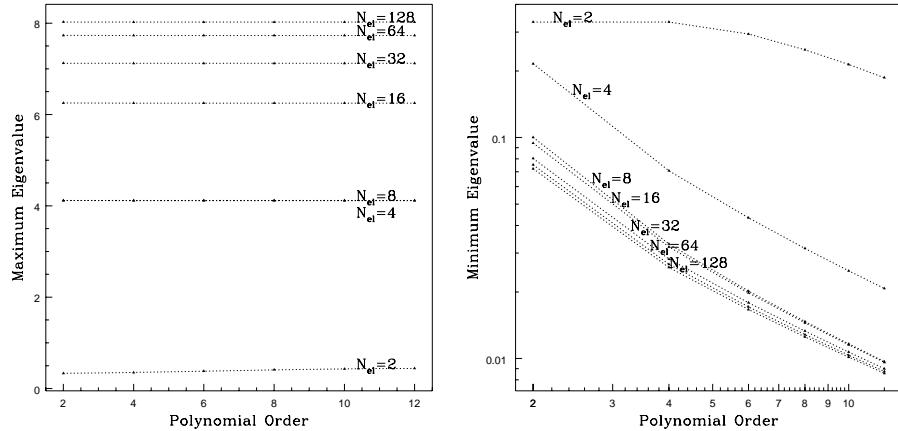


FIGURE 14. Maximum (left) and minimum (right) eigenvalue of the Schur complement of the discrete Laplacian operator with respect to the expansion order.

polynomial order P is demonstrated in figure 13(a), and with the number of elements in figure 13(b). The variation of the maximum and minimum eigenvalue with respect to the expansion order is shown in figure 14. The maximum eigenvalue is independent of the order P in accordance with the estimates in [3]. Also, the minimum eigenvalue varies as $\approx 1/P$ which is consistent with the theoretical upper bound estimate in [3] of $\log(P)/P$. For the range considered, these results also seem to indicate that the condition number grows at most linearly with the order P and slower than logarithmically with the number of elements N_{el} .

To get a better indication of the asymptotic behaviour with the polynomial order, P , we can consider the case of two elemental regions for the polynomial

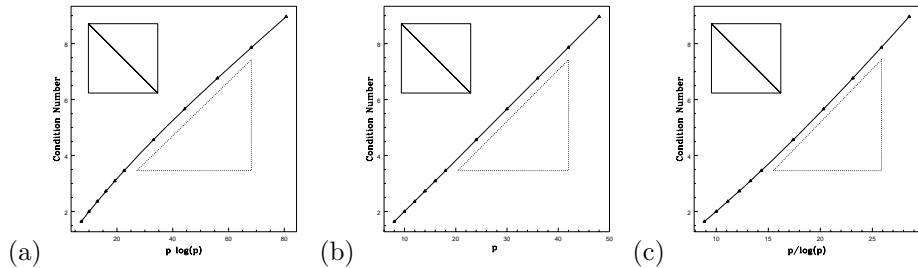


FIGURE 15. Condition number of the Schur complement of the discrete Laplacian operator for the $N_{el} = 2$ element domain plotted as a function of (a) $P \log(P)$, (b) P and (c) $P/\log(P)$.

range $8 \leq P \leq 48$ as shown in figure 15. In this test case we have imposed Dirichlet boundary conditions on all boundaries and so when we statically condense the system we are only left with the interior edge system. In figure 15 we see the condition number for this problem plotted against the functions (a) $P \log(P)$, (b) P and (c) $P/\log(P)$. As can be seen the condition number clearly grows at a slower rate than $P \log(P)$ but faster than $P/\log(P)$ which is consistent with the results of [4]. Although, not formally proven the asymptotic rate is most likely to scale with P . To extend this result to many subdomains the upper and lower bounds on the condition number $P \log(P)$ and $P/\log(P)$ should be scaled by a factor of $\log(P)^2$ [3]. Therefore, the asymptotic bound on κ_2 for a large P and N_{el} is

$$P \log(P) \leq \kappa_2 \leq P \log(P)^3.$$

However, when the number of elements, N_{el} , is not large a more conservative bound is

$$P/\log(P) \leq \kappa_2 \leq P \log(P)^3.$$

Since the upper bound is only realized for a large N_{el} the upper bound in these estimates is often observed to be very conservative. Furthermore, one would expect to observe a sharp upper bound of $P \log(P)^2$ based on the numerical behaviour shown in figure 15 for large P . These results may equally well be applied to the quadrilateral region which have similar edge support.

If the diagonal of the Schur complement is used as preconditioner, then the same scaling applies but the absolute magnitude of the condition number is approximately one order of magnitude less compared with the unpreconditioned case. This is not, however, true for the modal basis constructed in [16] based on the integrated Legendre polynomials rather than the $P_p^{1,1}(x)$ Jacobi polynomials. The difference between these two formulations in the quadrilateral case is the presence of a factor of P that provides the proper scaling and thus the similar growth in the unpreconditioned and diagonal-preconditioned case. On the other hand, if a block-diagonal preconditioner is used which is constructed based on blocks of edge-edge interactions, then the scaling changes substantially. Numerical experiments suggest a scaling of $(\log(P))^2$, in agreement with the estimates reported in [1] for a similar construction. The condition number also seems to be independent of the number of elements for $N_{el} \geq 100$, again in agreement with the estimates in [1].

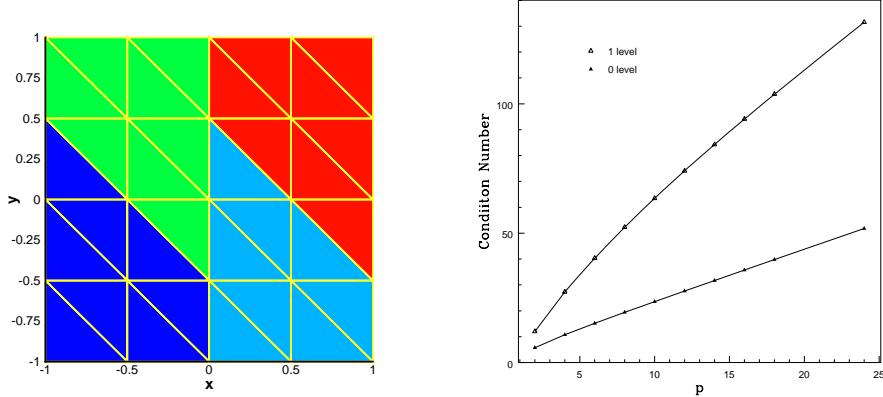


FIGURE 16. Using the $N_{el} = 32$ element mesh shown on the left the condition number of the standard Schur complement and the Schur complement after one level of decomposition were calculated and are shown on the right. The color in the left plot indicates the non-overlapping regions used for the extra level of decomposition.

As mentioned previously it can be shown [15] that for a symmetric positive definite matrix the maximum and minimum eigenvalue of the Schur complement are bounded by the maximum and minimum eigenvalue of the original matrix. Therefore, when using the multi-level Schur complement solver we know that the condition number of the inner-most Schur complement must be bounded by the standard Schur complement. This point is illustrated in figure 16 where we consider the condition number of the diagonally preconditioned standard Schur complement as well as the diagonally preconditioned Schur complement after one level of decomposition of a $N_{el} = 32$ elemental domain. The standard Schur complement contains information from all edges interior to the domain whereas in decomposing the system by one level all the edges within a shaded region are blocked together and statically condensed out of the system leaving only the edges along the interfaces between the shaded regions, see section 4.1.1. From the right hand plot in figure 16 we see that the condition number of the Schur complement after one level of decomposition is bounded by the condition number of the standard Schur complement. The effect of the extra level of decomposition is to reduce the slope of the curve although the condition number would appear still to be asymptotically growing with P .

The efficiency by which we can invert the *Helmholtz* matrix depends on the *combined* spectrum of the Laplacian matrix and the mass matrix. The eigen-spectrum of the Schur complement of the mass matrix has not been studied theoretically but numerical experiments with triangular elements suggest a similar dependence as the Laplacian matrix with respect to the number of elements N_{el} . However, with respect to order P its condition number grows much faster. In particular, for no-preconditioning we observed a growth $\kappa_2(\mathbf{M}) \propto P^{5/2}$; for diagonal-preconditioning $\kappa_2(\mathbf{M}) \propto P^{1.95}$; and for block-diagonal preconditioning as before we obtained $\kappa_2(\mathbf{M}) \propto P^{1.6}$. These are probably poly-logarithmic terms but we best-fitted experimental results to obtain these exponents.

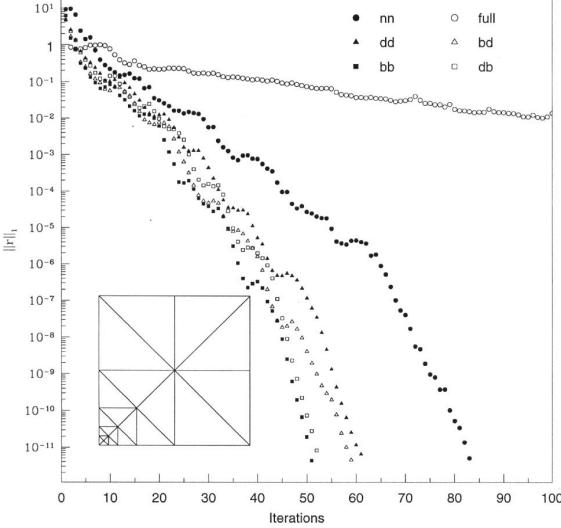


FIGURE 17. Convergence rate of a preconditioned conjugate gradient solver for a Schur complement of the Helmholtz matrix using the following preconditioners: (nn) vertex-non, edge-non; (dd) vertex-diagonal, edge-diagonal; (bd) vertex-diagonal, edge-block; (db) vertex-block, edge-diagonal; (bb) vertex-block, edge-block. Also shown for comparison is the solution for the full Helmholtz matrix (full).

In summary, it is possible to apply preconditioning techniques for inverting the Helmholtz matrix similar to preconditioners for the Laplacian and the mass matrix. The effect on the convergence rate of a preconditioned conjugate gradient solver for the Helmholtz equation with constant $\lambda = 1$ is shown in figure 17, which verifies the fast convergence for the Schur complement in contrast with the full discrete Laplacian.

In three dimensions, it is more difficult to establish estimates of the condition number. Some numerical experiments show that $\kappa_2(\mathbf{M}_S) \propto (\log(P))^8$ without any preconditioning, but they are inconclusive with respect to the dependence in terms of the number of elements. In [9] a polylogarithmic bound was found of the form

$$\kappa_2 \leq C(1 + \log(P))^2$$

which is independent on the number of elements. This estimate is valid for hexahedral elements but a similar bound was obtained in [2] for tetrahedral elements. The main idea is to use a wire basket preconditioner that is based on a new set of vertex and edge basis functions of “low energy”. These low energy functions with highly oscillatory traces on the wire basket decay much faster than the standard basis functions constructed using barycentric coordinates. An alternative approach that employs orthogonalisation of each vertex function with respect to functions of its three faces, and each edge function with respect to functions of its two faces, has been proposed in [7].

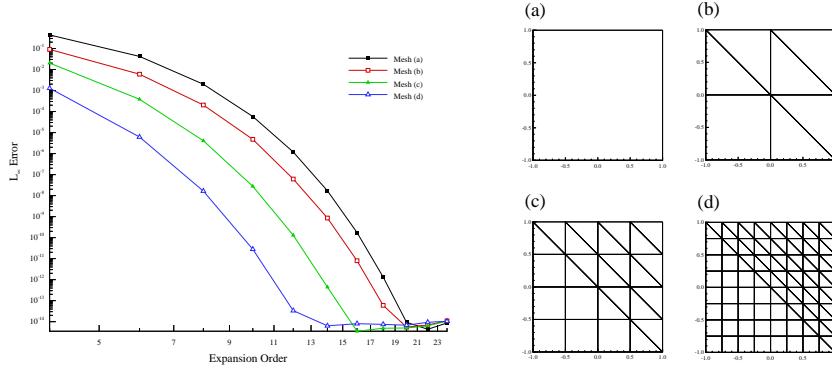


FIGURE 18. Convergence test for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, using quadrilaterals and triangles, with Dirichlet boundary conditions. The exact solution is $u = \sin(\pi x)\cos(\pi y)$ and forcing function $f = -(\lambda + 2\pi^2)\sin(\pi x)\cos(\pi y)$.

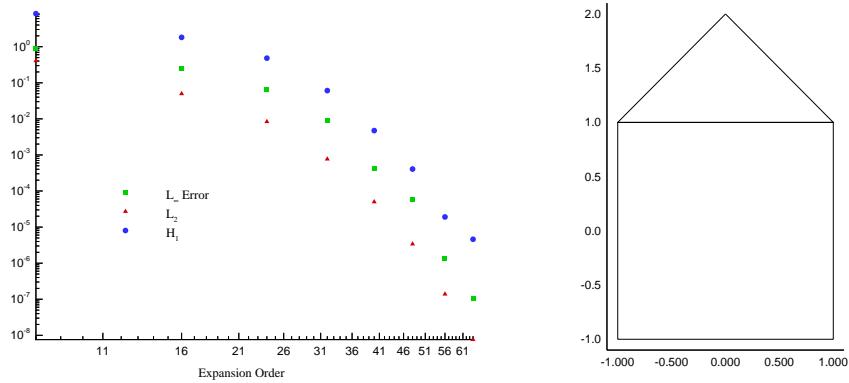


FIGURE 19. Convergence test for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, using a triangle and a quadrilateral, with Dirichlet boundary conditions. The exact solution is $u = \sin(\pi \cos(\pi r^2))$ and forcing function $f = -(\lambda + 4\pi^4 r^2 \sin(\pi r^2)^2) \sin(\pi(\cos(\pi r^2))) - 4\pi^2 (\pi r^2 \cos(\pi r^2) + \sin(\pi r^2)) \cos(\pi(\cos(\pi r^2)))$, where $r^2 = x^2 + y^2$.

5. Results

We now demonstrate that the method is stable up to high polynomial orders and works for complicated combinations of all the element types.

In figure 18 we demonstrate convergence to the exact solution with p -refinement (exponential rate) and h -refinement (algebraic rate) for the Helmholtz equation with $\lambda = 1$ for Dirichlet boundary conditions.

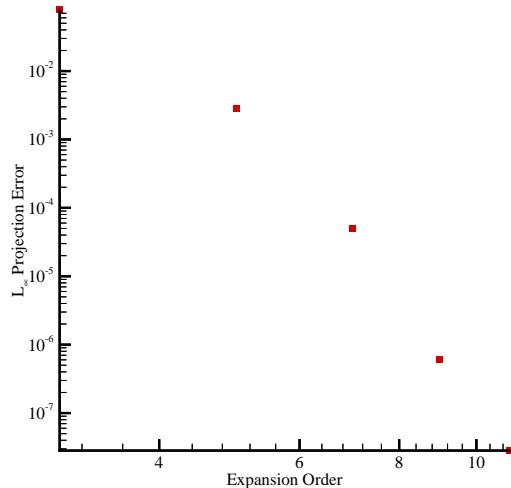
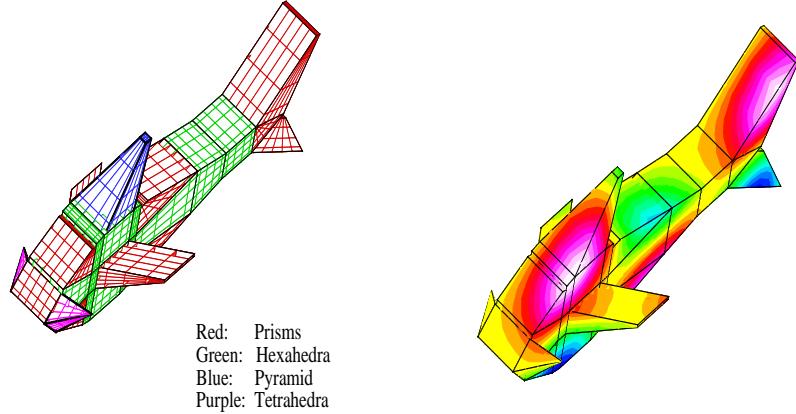


FIGURE 20. Convergence for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, with Dirichlet boundary conditions on a mesh of twenty six hybrid elements. The exact solution is $\sin(x)\sin(y)\sin(z)$.

In figure 19 we show p -type convergence for a more complicated exact solution. This example demonstrates that the method is stable to at least $P = 64$. This is much higher order than the one used in hp finite element method [16].

In figure 20 we solve the Helmholtz problem on a complicated 3D domain discretized using all types of elements, i.e. tetrahedra, hexahedra, prisms and pyramids. Exponential convergence is also verified for a smooth solution.

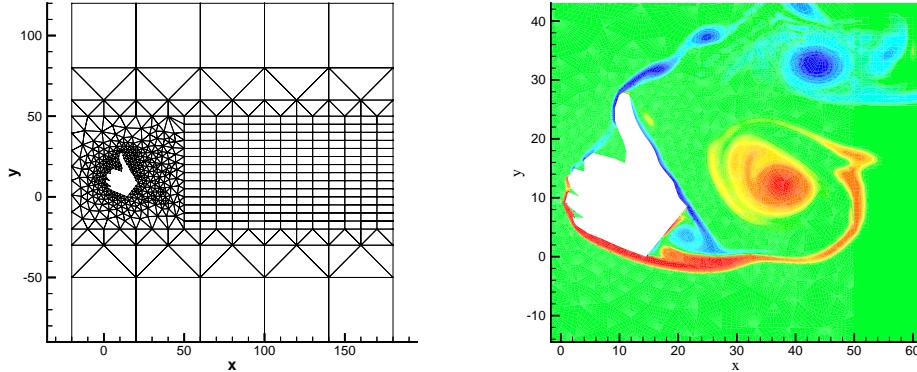


FIGURE 21. $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ simulations of incompressible flow on the above domain discretized with $N_{el} = 966$ elements (192 quadrilaterals and 774 triangles) and 6th order polynomial expansions on each element. The instantaneous vorticity field is shown on the right plot. The Reynolds number is approximately 1200.

Finally, in figure 21 we show a solution of the incompressible Navier-Stokes equations. The algorithm for triangular elements is described in detail in [13] but here we use a mix of triangles and quadrilateral elements, the former to handle the complex geometry and the latter to more efficiently fill the computational domain. The flow computed is start-up from zero initial conditions with uniform inflow at Reynolds number (based on the vertical projected length) approximately 1200.

Acknowledgements

We would like to thank Dr. Mario Casarin for many helpful discussions regarding the scaling of the condition number presented in this paper. This work was partially supported by AFOSR, DOE, ONR and NSF. Computations were performed at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, and at Maui High Performance Computing Center in Hawaii.

References

1. I. Babuška, A.W. Craig, J. Mandel, and J. Pitkaranta, *Efficient preconditioning for the p -version finite element method in two dimensions*, SIAM J. Numer. Anal. **28** (1991), 624–662.
2. I. Bica, *Iterative substructuring algorithms for the p -version finite element method for elliptic problems*, Ph.D. thesis, New York University, September 1997.
3. M.A. Casarin, *Schwarz preconditioners for spectral and mortar finite element methods with applications to incompressible fluids*, Ph.D. thesis, New York University, March 1996.
4. ———, *Diagonal edge preconditioners in p -version and spectral element methods*, SIAM J. Sci. Comp. **18** (1997), no. 2, 610–620.
5. Y. Kallinderis, A. Khawaja, and H. McMorris, *Hybrid prismatic/tetrahedral grid generation for complex geometries*, Tech. report, University of Texas at Austin, 1995.
6. G.E. Karniadakis and S.J. Sherwin, *Spectral/ hp element methods for cfd*, Oxford University Press, New York, 1998.
7. J. Mandel, *Two-level decomposition preconditioning for the p -version finite element method in three dimensions*, Int. J. Num. Meth. Engrg. **29** (1991), 1095–1108.

8. J.T. Oden, *Optimal hp-finite element methods*, Tech. Report TICOM Report 92-09, University of Texas at Austin, 1992.
9. L. Pavarino and O. Widlund, *A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions*, SIAM J. Numer. Anal. **33** (1996), 1303–1335.
10. S.J. Sherwin, *Hierarchical hp finite elements in hybrid domains*, Finite Elements in Analysis and Design **27** (1997), 109–119.
11. S.J. Sherwin, C. Evangelinos, H. Tufo, and G.E. Karniadakis, *Development of a parallel unstructured spectral/hp method for unsteady fluid dynamics*, Parallel CFD '97, 1997, Manchester, UK.
12. S.J. Sherwin and G.E. Karniadakis, *A new triangular and tetrahedral basis for high-order finite element methods*, Int. J. Num. Meth. Eng. **38** (1995), 3775.
13. ———, *A triangular spectral element method; applications to the incompressible Navier-Stokes equations*, Comp. Meth. Appl. Mech. Eng. **123** (1995), 189.
14. ———, *Tetrahedral hp finite elements: Algorithms and flow simulations*, J. Comp. Phys. **124** (1996), 14.
15. B. Smith, P. BJORSTAD, and W. GROPP, *Domain decomposition. parallel multilevel methods for elliptic differential equations*, Cambridge University Press, 1996.
16. B. Szabo and I. Babuška, *Finite Element Analysis*, John Wiley & Sons, 1991.
17. T.C.E. Warburton, *Spectral/hp element methods on polymorphic domains*, Ph.D. thesis, Brown University, 1998.

DEPARTMENT OF AERONAUTICS, IMPERIAL COLLEGE, PRINCE CONSORT ROAD, LONDON. SW7 2BY

CENTER FOR FLUID MECHANICS, DIVISION OF APPLIED MATHEMATICS, BROWN UNIVERSITY,
PROVIDENCE, R.I. 02912

CENTER FOR FLUID MECHANICS, DIVISION OF APPLIED MATHEMATICS, BROWN UNIVERSITY,
PROVIDENCE, R.I. 02912

Physical and Computational Domain Decompositions for Modeling Subsurface Flows

Mary F. Wheeler and Ivan Yotov

1. Introduction

Modeling of multiphase flow in permeable media plays a central role in subsurface environmental remediation as well as in problems associated with production of hydrocarbon energy from existing oil and gas fields. Numerical simulation is essential for risk assessment, cost reduction, and rational and efficient use of resources.

The contamination of groundwater is one of the most serious environmental problems facing the world. For example, more than 50% of drinking water in the United States comes from groundwater. More than 10,000 active military installations and over 6,200 closed installations in the United States require subsurface remediation. The process is difficult and extremely expensive and only now is technology emerging to cope with this severe and widespread problem. Hydrocarbons contribute almost two-thirds of the nation's energy supply. Moreover, recoverable reserves are being increased twice as fast by enhanced oil recovery techniques as by exploration.

Features that make the above problems difficult for numerical simulation include: multiple phases and chemical components, multi-scale heterogeneities, stiff gradients, irregular geometries with internal boundaries such as faults and layers, and multi-physics. Because of the uncertainty in the data, one frequently assumes stochastic coefficients and thus is forced to multiple realizations; therefore both computational efficiency and accuracy are crucial in the simulations. For efficiency, the future lies in developing parallel simulators which utilize domain decomposition algorithms.

One may ask what are the important aspects of parallel computation for these complex physical models. First, in all cases, one must be able to partition dynamically the geological domain based upon the physics of the model. Second, efficient distribution of the computations must be performed. Critical issues here are load balancing and minimal communication overhead. It is important to note that the two decompositions may be different.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 76S05.

This work was supported in part by the U.S. Department of Energy and the National Science Foundation.

In this paper we will discuss a novel numerical methodology for subsurface modeling based on multiblock domain decomposition formulations. Multiblock discretizations involve the introduction of special approximating spaces (mortars) on interfaces of adjacent subdomains. This paradigm is consistent with a physical/engineering description of the mathematical equations: that is, the equations hold with their usual meaning on the sub-domains, which have physically meaningful interface boundary conditions between them. The following features make the multiblock approach computationally attractive.

In many cases geometrically highly irregular domains can be described as unions of relatively simple blocks. Each block is independently covered by a relatively simple (e.g. logically rectangular) grid. The grids do not have to match on the interfaces between blocks. The local grid structure allows for more efficient and accurate discretization techniques to be employed. (For example, mixed finite element/finite volume methods are more accurate and efficient on structured than unstructured grids). Moreover, structured and unstructured grids could be coupled, if the geometry of a given block is very irregular.

Since the numerical grids may be non-matching across interfaces, they can be constructed to follow large scale geological features such as faults, heterogeneous layers, and other internal boundaries. This is critical for the accuracy of the numerical methods.

The multiblock approach allows for rigorous coupling of different physical processes, mathematical models, or discretization methods in different parts of the simulation domain (e.g., coupling underground with surface flow or coupling mixed finite element with standard finite element methods).

Dynamic grid adaptivity can be performed locally on each block. This is very convenient for the fast reconstruction of grids and calculation of stiffness matrices in time-dependent problems. Mortar degrees of freedom may also vary, providing an additional degree of adaptivity. For complex problems with multiscale heterogeneities and behavior, this approach provides a new mechanism for upscaling by computing an effective flow field without having to compute effective permeabilities. Moreover, the jump in fluxes along interfaces is a good indicator for the magnitude of the local discretization error.

The multiblock structure of the discrete systems of equations allows for efficient parallel domain decomposition solvers and preconditioners, which maximize data and computation locality, to be designed and applied. In addition, weighted space filling curve techniques provide efficient tools for distributing computations among processors. Near optimal load balancing and minimal communication overhead can be achieved, even for unstructured or dynamically adapted grids and computationally rough problems (problems with a nonuniform computational load) [26].

Mortar finite elements have been successfully applied for standard finite element and spectral finite element discretizations on non-matching grids (see, e.g. [9, 8]).

We have demonstrated in recent work that mortar domain decomposition is a viable approach for modeling subsurface flow and transport. Physical and mathematical considerations lead us to emphasize locally mass conservative schemes, in particular mixed finite element (finite volume) methods for subdomain discretizations. Theoretical and numerical results for single phase flow indicate multiblock mixed finite element methods are highly accurate (superconvergent) for both pressure and velocity [27, 1, 5, 7, 29]. A parallel non-overlapping domain decomposition implementation, based on a method originally proposed by Glowinski and

Wheeler [16, 13, 12], provides an efficient scalable solution technique [27]. Some efficient preconditioners have also been developed [18]. An extension of the method to a degenerate parabolic equation arising in two phase flow is presented in [28], where optimal convergence is shown.

In this paper we present a nonlinear domain decomposition algorithm for multiphase flow in porous media, based on mortar mixed finite element discretizations. The global discrete nonlinear system of equations is reduced to a nonlinear interface problem in the mortar space. The results demonstrate that this approach works very well for systems of transient highly non-linear differential equations.

The rest of the paper is organized as follows. In the next section we present a multiblock formulation and discretization for a two phase flow model. The domain decomposition algorithm is described in Section 3. Computational results, including some results on mortar adaptivity and upscaling are given in Section 4. We close in Section 5 with remarks on possible extensions and conclusions.

2. Multiblock formulation and discretization

To illustrate the numerical technique, we consider a two-phase flow model. In a multiblock formulation the domain $\Omega \subset \mathbf{R}^3$ is decomposed into a series of subdomains Ω_k , $k = 1, \dots, n_b$. Let $\Gamma_{kl} = \partial\Omega_k \cap \partial\Omega_l$ be the interface between Ω_k and Ω_l . We note that Γ_{kl} does not have to coincide with an edge (face) of either subdomain.

The governing mass conservation equations are imposed on each subdomain Ω_k :

$$(1) \quad \frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot \mathbf{U}_\alpha = q_\alpha,$$

where $\alpha = w$ (wetting), n (non-wetting) denotes the phase, S_α is the phase saturation, $\rho_\alpha = \rho_\alpha(P_\alpha)$ is the phase density, ϕ is the porosity, q_α is the source term, and

$$(2) \quad \mathbf{U}_\alpha = -\frac{k_\alpha(S_\alpha)K}{\mu_\alpha} \rho_\alpha (\nabla P_\alpha - \rho_\alpha g \nabla D)$$

is the Darcy velocity. Here P_α is the phase pressure, $k_\alpha(S_\alpha)$ is the phase relative permeability, μ_α is the phase viscosity, K is the rock permeability tensor, g is the gravitational constant, and D is the depth. On each interface Γ_{kl} the following physically meaningful continuity conditions are imposed:

$$(3) \quad P_\alpha|_{\Omega_k} = P_\alpha|_{\Omega_l},$$

$$(4) \quad [\mathbf{U}_\alpha \cdot \nu]_{kl} \equiv \mathbf{U}_\alpha|_{\Omega_k} \cdot \nu_k + \mathbf{U}_\alpha|_{\Omega_l} \cdot \nu_l = 0,$$

where ν_k denotes the outward unit normal vector on $\partial\Omega_k$. The above equations are coupled via the volume balance equation and the capillary pressure relation

$$(5) \quad S_w + S_n = 1, \quad p_c(S_w) = P_n - P_w,$$

which are imposed on each Ω_k and Γ_{kl} . We assume that no flow $\mathbf{U}_\alpha \cdot \nu = 0$ is imposed on $\partial\Omega$, although more general types of boundary conditions can also be treated.

2.1. Discretization spaces. It is important to choose properly the subdomain and interface discretization spaces in order to obtain a stable and accurate scheme. A variant of the mixed method, the expanded mixed method, has been developed for accurate and efficient treatment of irregular domains. The implementation and analysis of the method for single phase flow have been described in several previous works (see [6, 2, 3] for single block and [27, 5, 29] for multiblock domains). The original problem is transformed into a problem on a union of regular computational (reference) grids. The permeability after the mapping is usually a full tensor (except in some trivial cases). The mixed method could then be approximated by cell-centered finite differences for the pressure, which is an efficient and highly accurate scheme [6].

To simplify the presentation we will only describe here the rectangular reference case. For a definition of the spaces on logically rectangular and triangular grids, we refer to [2] (also see [24, 10]). Let us denote the rectangular partition of Ω_k by \mathcal{T}_{h_k} , where h_k is associated with the size of the elements. The lowest order Raviart-Thomas spaces RT_0 [23] are defined on \mathcal{T}_{h_k} by

$$\begin{aligned} \tilde{\mathbf{V}}_{h_k} &= \left\{ \mathbf{v} = (v_1, v_2, v_3) : \mathbf{v}|_E = (\alpha_1 x_1 + \beta_1, \alpha_2 x_2 + \beta_2, \alpha_3 x_3 + \beta_3)^T : \right. \\ &\quad \left. \alpha_l, \beta_l \in \mathbf{R} \text{ for all } E \in \mathcal{T}_{h_k}, \right. \\ &\quad \left. \text{and each } v_l \text{ is continuous in the } l\text{th coordinate direction} \right\}, \\ \mathbf{V}_{h_k} &= \left\{ \mathbf{v} \in \tilde{\mathbf{V}}_{h_k} : \mathbf{v} \cdot \nu_k = 0 \text{ on } \partial\Omega_k \cap \partial\Omega \right\} \\ W_{h_k} &= \left\{ w : w|_E = \alpha : \alpha \in \mathbf{R} \text{ for all } E \in \mathcal{T}_{h_k} \right\}. \end{aligned}$$

To impose the interface matching condition (3)–(4) we introduce a Lagrange multiplier or mortar finite element space $M_{h_{kl}}$ defined on a rectangular grid $\mathcal{T}_{h_{kl}}$ on Γ_{kl} , where h_{kl} is associated with the size of the elements in $\mathcal{T}_{h_{kl}}$. In this space we approximate the interface pressures and saturations, and impose weakly normal continuity of fluxes.

If the subdomain grids adjacent to Γ_{kl} match, we take $\mathcal{T}_{h_{kl}}$ to be the trace of the subdomain grids and define the matching mortar space by

$$M_{h_{kl}}^m = \left\{ \mu : \mu|_e = \alpha : \alpha \in \mathbf{R}, \text{ for all } e \in \mathcal{T}_{h_{kl}} \right\}.$$

If the grids adjacent to Γ_{kl} are non-matching, the interface grid need not match either of them. Later we impose a mild condition on $\mathcal{T}_{h_{kl}}$ to guarantee solvability of the numerical scheme. We define our non-matching mortar space on an element $e \in \mathcal{T}_{h_{kl}}$ by

$$M_h^n(e) = \left\{ \alpha\xi_1\xi_2 + \beta\xi_1 + \gamma\xi_2 + \delta : \alpha, \beta, \gamma, \delta \in \mathbf{R} \right\},$$

where ξ_l are the coordinate variables on e . Then, for each Γ_{kl} , we give two possibilities for the non-matching mortar space, a discontinuous and a continuous version, as

$$\begin{aligned} M_{h_{kl}}^{n,d} &= \left\{ \mu : \mu|_e \in M_h^n(e) \text{ for all } e \in \mathcal{T}_{h_{kl}} \right\}, \\ M_{h_{kl}}^{n,c} &= \left\{ \mu : \mu|_e \in M_h^n(e) \text{ for all } e \in \mathcal{T}_{h_{kl}}, \mu \text{ is continuous on } \Gamma_{kl} \right\}. \end{aligned}$$

We denote by $M_{h_{kl}}$ any choice of $M_{h_{kl}}^{n,d}$, $M_{h_{kl}}^{n,c}$, or $M_{h_{kl}}^m$ (on matching interfaces).

REMARK 1. The usual piece-wise constant Lagrange multiplier space for RT_0 is not a good choice in the case of non-matching grids, since it only provides $O(1)$ approximation on the interfaces and a suboptimal global convergence. With the

above choice for mortar space, optimal convergence and, in some cases, superconvergence is recovered for both pressure and velocity (see [27, 1] for single phase flow and [28] for two phase flow).

2.2. The expanded mortar mixed finite element method. Following [6], let, for $\alpha = w, n$,

$$\tilde{\mathbf{U}}_\alpha = -\nabla P_\alpha.$$

Then

$$\mathbf{U}_\alpha = -\frac{k_\alpha(S_\alpha)K}{\mu_\alpha} \rho_\alpha (\tilde{\mathbf{U}}_\alpha - \rho_\alpha g \nabla D).$$

Before formulating the method, we note that two of the unknowns can be eliminated using relations (5). Therefore the primary variables can be chosen to be one pressure and one saturation which we denote by P and S .

Let $0 = t_0 < t_1 < t_2 < \dots$, let $\Delta t^n = t_n - t_{n-1}$, and let $f^n = f(t_n)$.

In the backward Euler multiblock expanded mixed finite element approximation of (1)-(5) we seek, for $1 \leq k < l \leq n_b$ and $n = 1, 2, 3, \dots$, $\mathbf{U}_{h,\alpha}|_{\Omega_k} \in \mathbf{V}_{h_k}$, $\tilde{\mathbf{U}}_{h,\alpha}|_{\Omega_k} \in \tilde{\mathbf{V}}_{h_k}$, $P_h^n|_{\Omega_k} \in W_{h_k}$, $S_h^n|_{\Omega_k} \in W_{h_k}$, $\bar{P}_h^n|_{\Gamma_{kl}} \in M_{h_{kl}}$, and $\bar{S}_h^n|_{\Gamma_{kl}} \in M_{h_{kl}}$ such that, for $\alpha = w$ and n ,

$$(6) \quad \int_{\Omega_k} \frac{S_{h,\alpha}^n - S_{h,\alpha}^{n-1}}{\Delta t^n} w \, dx + \int_{\Omega_k} \nabla \cdot \mathbf{U}_{h,\alpha}^n w \, dx = \int_{\Omega_k} q_\alpha w \, dx, \quad w \in W_{h_k},$$

$$(7) \quad \int_{\Omega_k} \tilde{\mathbf{U}}_{h,\alpha}^n \cdot \mathbf{v} \, dx = \int_{\Omega_k} P_{h,\alpha}^n \nabla \cdot \mathbf{v} \, dx - \int_{\partial\Omega_k \setminus \partial\Omega} \bar{P}_{h,\alpha}^n \mathbf{v} \cdot \nu_k \, d\sigma, \quad \mathbf{v} \in \mathbf{V}_{h_k},$$

$$(8) \quad \int_{\Omega_k} \mathbf{U}_{h,\alpha}^n \cdot \tilde{\mathbf{v}} \, dx = \int_{\Omega_k} \frac{k_{h,\alpha}^n K}{\mu_{h,\alpha}} \rho_{h,\alpha}^n (\tilde{\mathbf{U}}_{h,\alpha}^n - \rho_{h,\alpha}^n g \nabla D) \cdot \tilde{\mathbf{v}} \, dx, \quad \tilde{\mathbf{v}} \in \tilde{\mathbf{V}}_{h_k},$$

$$(9) \quad \int_{\Gamma_{kl}} [\mathbf{U}_{h,\alpha}^n \cdot \nu]_{kl} \mu \, d\sigma = 0, \quad \mu \in M_{h_{kl}}.$$

Here $k_{h,\alpha}^n$ and $\rho_{h,\alpha}^n \in W_{h_k}$ are given functions of the subdomain primary variables P_h^n and S_h^n . The mortar functions \bar{P}_h^n can be computed using (5), given the mortar primary variables \bar{P}_h^n and \bar{S}_h^n .

REMARK 2. Introducing the pressure gradients $\tilde{\mathbf{U}}_\alpha$ in the expanded mixed method allows for proper handling of the degenerate (for $S_\alpha = 0$) relative permeability $k_\alpha(S_\alpha)$ in (7)–(8). It also allows, even for a full permeability tensor K , to accurately approximate the mixed method on each subdomain by cell-centered finite differences for P_h and S_h . This is achieved by approximating the vector integrals in (7) and (8) by a trapezoidal quadrature rule and eliminating $\tilde{\mathbf{U}}_{h,\alpha}$ and $\mathbf{U}_{h,\alpha}$ from the system [6, 2, 3].

REMARK 3. A necessary condition for solvability of the scheme is that, for any $\phi \in M_{h_{kl}}$,

$$(10) \quad Q_{h,k}\phi = Q_{h,l}\phi = 0 \Rightarrow \phi = 0,$$

where $Q_{h,k}$ is the L^2 -projection onto $\mathbf{V}_{h_k} \cdot \nu_k$. This is not a very restrictive condition and requires that the mortar grid is not too fine compared to the subdomain grids. One choice that satisfies this condition for both continuous and discontinuous mortars is to take the trace of either subdomain grid and coarsen it by two in each direction (see [27, 1] for details).

3. Domain decomposition

To solve the discrete system (6)–(9) on each time step, we reduce it to an interface problem in the mortar space. This approach is based on a domain decomposition algorithm for single phase flow developed originally for conforming grids [16], and later generalized to non-matching grids coupled with mortars [27].

3.1. Interface formulation.

Let

$$M_h = \bigoplus_{1 \leq k < l \leq n_b} M_{h_{kl}}$$

denote the mortar space on $\Gamma = \cup_{1 \leq k < l \leq n_b} \Gamma_{kl}$ and let $\mathbf{M}_h = M_h \times M_h$. We define a non-linear interface functional $B^n : \mathbf{M}_h \times \mathbf{M}_h \rightarrow \mathbf{R}$ as follows. For $\psi = (\bar{P}_h^n, \bar{S}_h^n)^T \in \mathbf{M}_h$ and $\mu = (\mu_w, \mu_n) \in \mathbf{M}_h$, let

$$B^n(\psi, \mu) = \sum_{1 \leq k < l \leq n_b} \int_{\Gamma_{kl}} ([\mathbf{U}_{h,w}^n(\psi) \cdot \nu]_{kl} \mu_w + [\mathbf{U}_{h,n}^n(\psi) \cdot \nu]_{kl} \mu_n) d\sigma,$$

where $(S_h^n(\psi), \mathbf{U}_{h,\alpha}^n(\psi))$ are solutions to the series of subdomain problems (6)–(8) with boundary data $\bar{P}_{h,\alpha}^n(\psi)$.

Define a non-linear interface operator $\mathcal{B}^n : \mathbf{M}_h \rightarrow \mathbf{M}_h$ by

$$\langle \mathcal{B}^n \psi, \mu \rangle = B^n(\psi, \mu), \quad \forall \mu \in \mathbf{M}_h,$$

where $\langle \cdot, \cdot \rangle$ is the L^2 -inner product in \mathbf{M}_h . It is now easy to see that the solution to (6)–(9) equals $(\psi, S_h^n(\psi), \mathbf{U}_{h,\alpha}^n(\psi))$, where $\psi \in \mathbf{M}_h$ solves

$$(11) \quad \mathcal{B}^n(\psi) = 0.$$

3.2. Iterative solution of the interface problem. We solve the system of nonlinear equations on the interface (11) by an inexact Newton method. Each Newton step s is computed by a forward difference GMRES iteration for solving $\mathcal{B}'(\psi)s = -\mathcal{B}(\psi)$ (we omit superscript n for simplicity). On each GMRES iteration the action of the Jacobian $\mathcal{B}'(\psi)$ on a vector μ is approximated by the forward difference

$$D_\delta \mathcal{B}(\psi : \mu) = \begin{cases} 0, & \mu = 0, \\ \|\mu\| \frac{\mathcal{B}(\psi + \delta \|\psi\| \mu / \|\mu\|) - \mathcal{B}(\psi)}{\delta \|\psi\|}, & \mu \neq 0, \psi \neq 0, \\ \|\mu\| \frac{\mathcal{B}(\delta \mu / \|\mu\|) - \mathcal{B}(\psi)}{\delta}, & \mu \neq 0, \psi = 0. \end{cases}$$

We take $\delta = \sqrt{\epsilon}$, where ϵ is the nonlinear tolerance for evaluation of \mathcal{B} . The inexact Newton-GMRES algorithm is described in details in [17].

Note that each GMRES iteration only requires one evaluation of the nonlinear operator \mathcal{B} . The evaluation of \mathcal{B} involves solving subdomain problems (6)–(8) in parallel and two inexpensive projection steps - from the mortar grid onto the local subdomain grids and from the local grids onto the mortar grid. Since each block can be distributed among a number of processors, the subdomain solvers are parallel themselves. This two level parallelism is needed to account for both the physical and the computational domain decomposition. The subdomain problems are also nonlinear and are solved by a preconditioned Newton-Krylov solver (see [14] for a detailed description). We must note that, since the perturbation δ is very small, the subdomain solution with boundary data ψ is a very good initial guess for solving

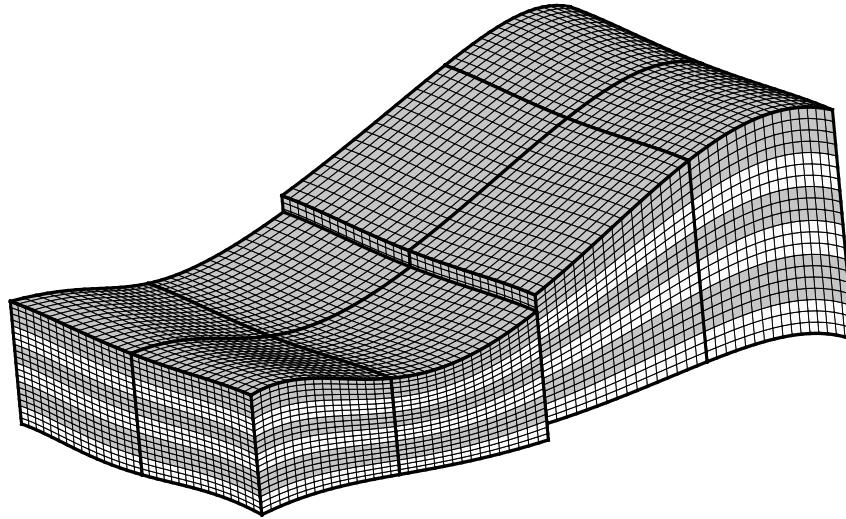


FIGURE 1. Geological layers and numerical grids. The dark layers (400 md) are eight times more permeable than the light layers.

subdomain problems with boundary data $\psi + \delta||\psi||\mu/||\mu||$. As a result it usually takes only one nonlinear subdomain iteration to evaluate $\mathcal{B}(\psi + \delta||\psi||\mu/||\mu||)$.

4. Computational results

In this section we present numerical results illustrating the application of the method described in the previous two sections to modeling two phase subsurface flow. We also give some results on adapting mortar degrees of freedom and its relation to upscaling in the case of single phase flow.

4.1. A two phase flow simulation. The methodology described above has been implemented in a parallel implicit multiblock two phase flow simulator UT-MB [25, 21]. The simulator is built on top of an object oriented parallel computational infrastructure [22], which is based on DAGH (Distributed Adaptive Grid Hierarchy) library [20].

In this example we present the results of a two phase oil-water flow simulation in a faulted heterogeneous irregularly shaped multiblock reservoir. A fault cuts through the middle of the domain and divides it into two blocks. The curvilinear numerical grids follow the geological layers and are non-matching across the fault (see Figure 1). Each block is covered by a $32 \times 32 \times 20$ grid. The simulation was done on eight processors on IBM SP2, each block distributed among four processors. Oil concentration contours after 281 days of displacement (water is injected at the right front corner and producer is placed at the left back corner) are given on Figure 2.

4.2. Mortar adaptivity and upscaling. Adapting mortar degrees of freedom may result in substantial reduction of the cost for solving the interface problem. Note that solvability condition (10) does not prevent from using mortar grids much coarser than the subdomain grids. One must expect, however, certain loss of accuracy with coarsening the interface grids. In the following example we study how

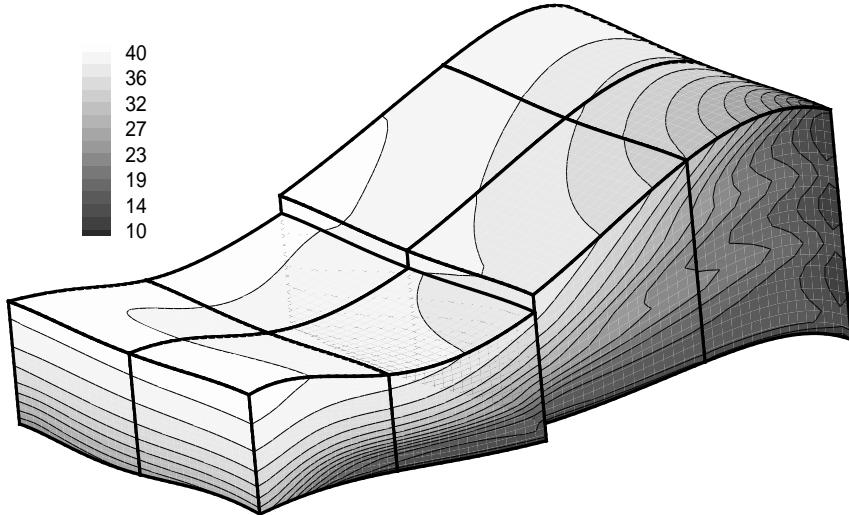


FIGURE 2. Oil concentration contours at 281 days.

reduction of mortar degrees of freedom affects the number of interface iterations and the flux discretization error on the interface. Similar ideas have been explored by Dorr in [15]. We solve a single phase flow problem on a $32 \times 32 \times 32$ domain with a highly correlated log-normal permeability field and one injection and three production wells at the corners. A $2 \times 2 \times 2$ domain decomposition is employed. This example suites well the purpose of our study, due to the large heterogeneities and substantial flow through all interfaces. The results of the experiment are shown in Figure 3. The traces of subdomain grids on each interface are 16×16 and having 256 mortar degrees of freedom is equivalent to exact matching of the fluxes. We report the number of conjugate gradient iterations (no preconditioning) and relative flux L^2 -error on the interface for several levels of coarsening the mortar grids and for three different types of mortars. We first note that the error for the piecewise constant mortars grows very rapidly and indicates that this is not a good choice. This is consistent with our theoretical results (see Remark 1). The two bilinear cases behave similarly, although the continuous case performs somewhat better. We observe that in this case, the number of mortar degrees of freedom, and consequently the number of interface iterations, can be reduced by a factor of two, with the relative flux error still being under ten percent. Moreover, the global relative error is even smaller, as the solution away from the interfaces is not affected as much.

The reduction of mortar degrees of freedom can be viewed as an upscaling procedure. Standard upscaling techniques compute effective permeabilities on coarse grids. It is usually difficult to estimate the error associated with the upscaling process. Here we compute, in a sense, an effective flow field and the flux jump is a good indication for the numerical error.

If only a single bilinear mortar is used on each interface, we have a two scale problem, where the solution is computed locally on the fine scale and fluxes match on the coarse (subdomain) scale. One can view the solution as a sum of a coarse

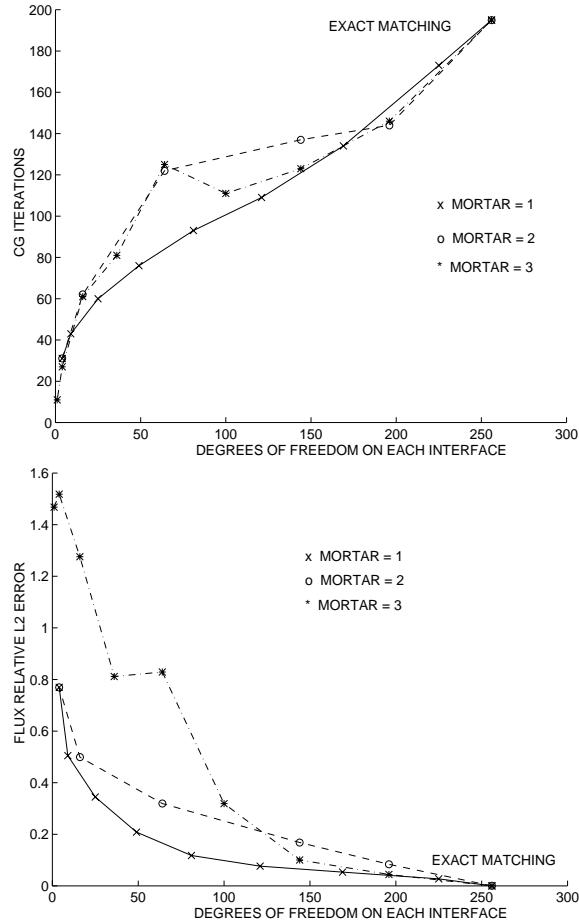


FIGURE 3. Dependence of interface iterations and error on number of interface degrees of freedom; mortar 1—continuous piecewise bilinears, mortar 2—discontinuous piecewise bilinears, mortar 3—piecewise constants.

grid solution and a local fine grid correction, which is similar to the approaches taken in [4, 19]. In the following example, also considered in [4], we solve the single phase flow equation with a log-normal permeability field originally presented in [11]. As can be seen in Figure 4, the solution on a fine 32×32 grid is very similar to the solution obtained by matching fluxes on a coarse 4×4 grid using a single linear mortar on each interface. We should note that a similar procedure using constant instead of linear mortars produced highly inaccurate results.

5. Conclusions

In this paper we considered two levels of domain decompositions - physical and computational. It is important to first decompose the physical problem with appropriate hierarchical models (geometry, geology, chemistry/physics) and then efficiently decompose the computations on a parallel machine.

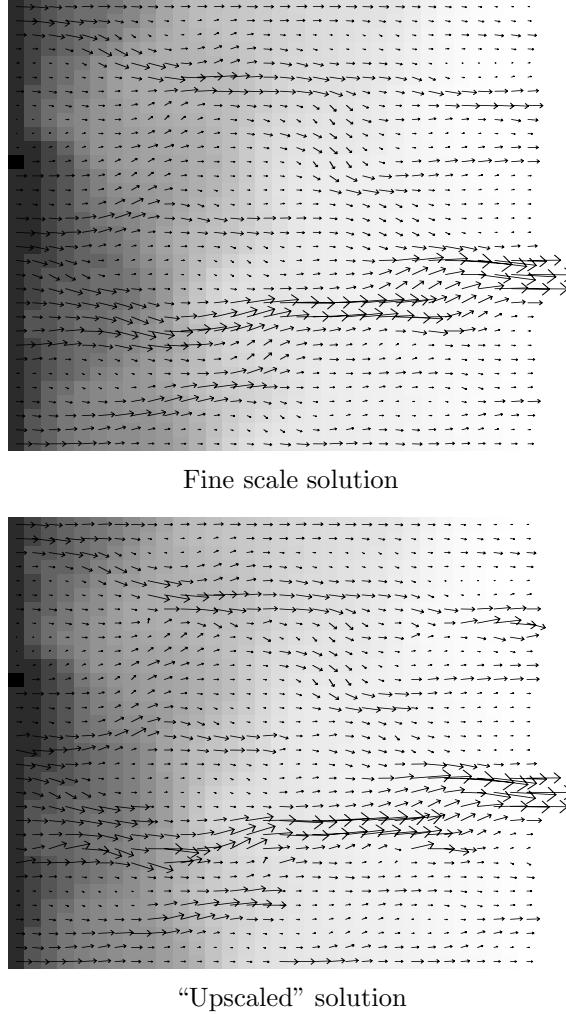


FIGURE 4. Computed pressure (shade) and velocity (arrows) field for the two scale example.

We have introduced new mortar spaces which provide an accurate and efficient basis for discretizations on non-matching grids, hierarchical domain decomposition, and solvers. In addition, this approach allows the coupling of multiphysics, multi-numerics, and multiscales.

We have demonstrated the applicability of these mortar space decompositions to two phase flow in permeable media. Further computational experiments have shown the computational cost can be reduced substantially by interface adaptivity, which is related to upscaling.

Our current research involves extensions of these techniques to three flowing phases and multiple solid phases, as well as coupling of fully implicit and various time-splitting schemes, as shown in Figure 5.

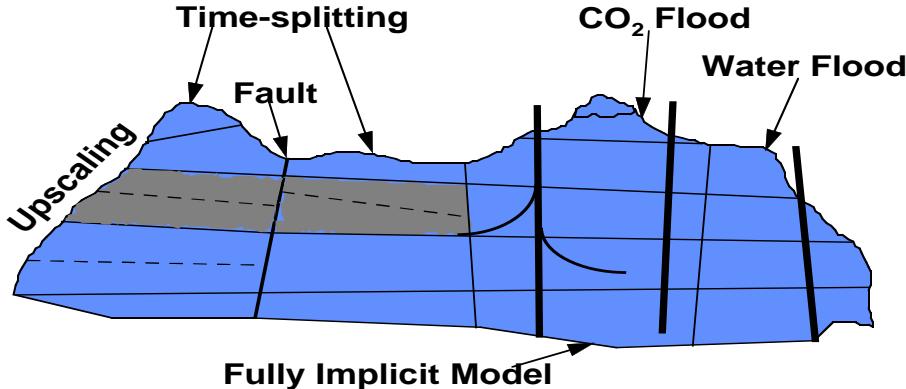


FIGURE 5. Multiphysics, multi-numerical models, complex geology, and upscaling.

References

1. T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, *Mixed finite element methods on non-matching multiblock grids*, Tech. Report TICAM 96-50, Texas Inst. Comp. Appl. Math., University of Texas at Austin, 1996, submitted to SIAM J. Num. Anal.
2. T. Arbogast, C. N. Dawson, P. T. Keenan, M. F. Wheeler, and I. Yotov, *Enhanced cell-centered finite differences for elliptic equations on general geometry*, SIAM J. Sci. Comp. **19** (1998), no. 2, 404–425.
3. T. Arbogast, P. T. Keenan, M. F. Wheeler, and I. Yotov, *Logically rectangular mixed methods for Darcy flow on general geometry*, Thirteenth SPE Symposium on Reservoir Simulation, San Antonio, Texas, Society of Petroleum Engineers, Feb. 1995, SPE 29099, pp. 51–59.
4. T. Arbogast, S. Minkoff, and P. Keenan, *An operator-based approach to upscaling the pressure equation*, in preparation.
5. T. Arbogast, M. F. Wheeler, and I. Yotov, *Logically rectangular mixed methods for flow in irregular, heterogeneous domains*, Computational Methods in Water Resources XI (A. A. Aldama et al., eds.), Computational Mechanics Publications, Southampton, 1996, pp. 621–628.
6. ———, *Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences*, SIAM J. Numer. Anal. **34** (1997), no. 2, 828–852.
7. T. Arbogast and I. Yotov, *A non-mortar mixed finite element method for elliptic problems on non-matching multiblock grids*, Comput. Meth. Appl. Mech. Eng. **149** (1997), 255–265.
8. F. Ben Belgacem and Y. Maday, *The mortar element method for three-dimensional finite elements*, RAIRO Mod. Math. Anal. Num. **31** (1997), no. 2, 289–302.
9. C. Bernardi, Y. Maday, and A. T. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, Nonlinear partial differential equations and their applications (H. Brezis and J. L. Lions, eds.), Longman Scientific & Technical, UK, 1994.
10. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, 1991.
11. M. A. Christie, M. Mansfield, P. R. King, J. W. Barker, and I. D. Culverwell, *A renormalization-based upscaling technique for WAG floods in heterogeneous reservoirs*, Expanded Abstracts, Society of Petroleum Engineers, 1995, SPE 29127, pp. 353–361.
12. L. C. Cowsar, J. Mandel, and M. F. Wheeler, *Balancing domain decomposition for mixed finite elements*, Math. Comp. **64** (1995), 989–1015.

13. L. C. Cowsar and M. F. Wheeler, *Parallel domain decomposition method for mixed finite elements for elliptic partial differential equations*, Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, Y. Kuznetsov, G. Meurant, J. Periaux, and O. Widlund, eds.), SIAM, Philadelphia, 1991.
14. C. N. Dawson, H. Klie, C. San Soucie, and M. F. Wheeler, *A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver*, Comp. Geosciences (1998).
15. M. R. Dorr, *On the discretization of interdomain coupling in elliptic boundary value problems*, Second International Symposium on Domain Decomposition Methods (T. F. Chan et al., eds.), SIAM, Philadelphia, 1989, pp. 17–37.
16. R. Glowinski and M. F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux, eds.), SIAM, Philadelphia, 1988, pp. 144–172.
17. C. T. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, 1995.
18. Y. A. Kuznetsov and M. F. Wheeler, *Optimal order substructuring preconditioners for mixed finite element methods on non-matching grids*, East-West J. Numer. Math. **3** (1995), no. 2, 127–143.
19. N. Moes, J. T. Oden, and K. Vemaganti, *A two-scale strategy and a posteriori error estimation for modeling heterogeneous structures*, On new advances in adaptive computational methods in mechanics, Elsevier, 1998, To appear.
20. M. Parashar and J. C. Browne, *An infrastructure for parallel adaptive mesh-refinement techniques*, Tech. report, Department of Computer Science, University of Texas at Austin, 1995.
21. M. Parashar, J. A. Wheeler, G. Pope, K. Wang, and P. Wang, *A new generation eos compositional reservoir simulator. part II: Framework and multiprocessing*, Fourteenth SPE Symposium on Reservoir Simulation, Dallas, Texas, Society of Petroleum Engineers, June 1997, pp. 31–38.
22. M. Parashar and I. Yotov, *Hybrid parallelization for multi-block multi-physics applications*, in preparation.
23. R. A. Raviart and J. M. Thomas, *A mixed finite element method for 2nd order elliptic problems*, Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics, vol. 606, Springer-Verlag, New York, 1977, pp. 292–315.
24. J. M. Thomas, *These de doctorat d'état*, à l'Université Pierre et Marie Curie, 1977.
25. P. Wang, I. Yotov, M. F. Wheeler, T. Arbogast, C. N. Dawson, M. Parashar, and K. Sepehrnoori, *A new generation eos compositional reservoir simulator. part I: Formulation and discretization*, Fourteenth SPE Symposium on Reservoir Simulation, Dallas, Texas, Society of Petroleum Engineers, June 1997, pp. 55–64.
26. M. F. Wheeler, C. N. Dawson, S. Chippada, H. C. Edwards, and M. L. Martinez, *Surface flow modeling of bays, estuaries and coastal oceans*, Parallel Computing Research **4** (1996), no. 3, 8–9.
27. I. Yotov, *Mixed finite element methods for flow in porous media*, Ph.D. thesis, Rice University, Houston, Texas, 1996, TR96-09, Dept. Comp. Appl. Math., Rice University and TICAM report 96-23, University of Texas at Austin.
28. ———, *A mixed finite element discretization on non-matching multiblock grids for a degenerate parabolic equation arising in porous media flow*, East-West J. Numer. Math. **5** (1997), no. 3, 211–230.
29. ———, *Mortar mixed finite element methods on irregular multiblock domains*, Third IMACS International Symposium on Iterative Methods in Scientific Computation (B. Chen, T. Mathew, and J. Wang, eds.), Academic Press, 1997, To appear.

TEXAS INSTITUTE FOR COMPUTATIONAL AND APPLIED MATHEMATICS, DEPARTMENT OF AEROSPACE ENGINEERING & ENGINEERING MECHANICS, AND DEPARTMENT OF PETROLEUM AND GEOSYSTEMS ENGINEERING, THE UNIVERSITY OF TEXAS AT AUSTIN, AUSTIN, TX 78712

E-mail address: mfw@ticam.utexas.edu

TEXAS INSTITUTE FOR COMPUTATIONAL AND APPLIED MATHEMATICS, THE UNIVERSITY OF TEXAS AT AUSTIN, AUSTIN, TX 78712

E-mail address: yotov@ticam.utexas.edu

Part 2

Algorithms

Nonoverlapping Domain Decomposition Algorithms for the p -version Finite Element Method for Elliptic Problems

Ion Bică

1. Introduction

The nonoverlapping domain decomposition methods form a class of domain decomposition methods, for which the information exchange between neighboring subdomains is limited to the variables directly associated with the interface, i.e. those common to more than one subregion. Our objective is to design algorithms in $3D$ for which we can find an upper bound on the *condition number* κ of the preconditioned linear system, which is independent of the number of subdomains and grows slowly with p . Here, p is the maximum degree of the polynomials used in the p -version finite element discretization of the continuous problem. In this paper, we survey some of the results obtained in [2].

Iterative substructuring methods for the h -version finite element, $2D$ p -version, and spectral elements have been previously developed and analyzed by several authors [3, 4], [6], [1], [13, 14], [11], [5], and [7, 8, 9].

However, some very real difficulties remained when the extension of these methods and their analysis to the $3D$ p -version finite element method were attempted, such as a lack of extension theorems for polynomials. The corresponding results are well known for Sobolev spaces, but their extension to finite element spaces is quite intricate. In our technical work, we use and further develop extension theorems for polynomials given in [1], [12], and [10] in order to prove the following bound on the condition number of our algorithm:

$$(1) \quad \kappa \leq C(1 + \log p)^4.$$

We believe that two logs can be dropped and a bound, similar to the ones in [3, 4], [6], and [13, 14], can be obtained.

In Section 2, we describe the model problem we are solving and the basis functions of the finite element space which are best suited for our algorithm. Section 3 contains a brief description of the preconditioner on which the algorithm is based. In Section 4, we compute the local and global condition numbers of our algorithm and make specific recommendations on the best choice of preconditioners.

1991 *Mathematics Subject Classification*. Primary 65N30; Secondary 41A10, 65N35, 65N55.

This work was supported in part by the National Science Foundation under Grants NSF-CCR-9503408 and in part by the U.S. Department of Energy under contract DE-FG02-92ER25127.

2. Continuous and discrete problems

We consider the following problem formulated variationally: Find $u \in V$ such that

$$(2) \quad a(u, v) = \int_{\Omega} \rho(x) \nabla u \nabla v \, dx = f(v) \quad \forall v \in V.$$

Here, V is a subspace of $H^1(\Omega)$, determined by boundary conditions, Ω is a polyhedral region triangulated with tetrahedra $\bar{\Omega}_i$, $\bar{\Omega} = \cup \bar{\Omega}_i$. We denote by Γ the interface between subdomains, $\Gamma = \cup \partial \Omega_i \setminus \partial \Omega$. We assume that the boundary conditions are of the same type within each face of any tetrahedron that is part of the boundary. The coefficient $\rho(x) > 0$ can be discontinuous across the interface between the subdomains, but varies only moderately within each Ω_i . Without further decreasing the generality, we assume $\rho(x) = \rho_i$ on Ω_i . The bound (1) holds for arbitrary jumps in ρ_i .

We discretize the problem by the p -version finite element method. The finite element space V^p consists of the continuous functions on Ω which are polynomials of total degree p in each Ω_i . We end up with a system $Kx = b$, where the stiffness matrix K is built from local stiffness matrices, by subassembly. We will now define the basis functions on a reference tetrahedron Ω_{ref} . There are many ways to do so, and a proper choice is the key to obtaining an efficient iterative method. We present here only a general description. We distinguish between four types of basis functions, associated with the vertices, edges, faces, and interior of Ω_{ref} .

1. A *vertex basis function* has value one at a vertex and vanishes on the face opposite to that vertex. There is only one vertex function per vertex.
2. An *edge basis function* vanishes on the two faces which do not share the edge. The traces of the edge functions associated with the same edge are chosen to be linearly independent on that edge. There are $p - 1$ such functions per edge.
3. A *face basis function* vanishes on the other three faces. The traces of the face functions associated with the same face are chosen to be linearly independent on that face. There are $(p - 1)(p - 2)/2$ such functions per face.
4. An *interior basis function* vanishes on $\partial \Omega_{ref}$. There are $(p-1)(p-2)(p-3)/6$ interior functions and they are linearly independent.

The total number of vertex, edge, face, and interior functions is $(p+1)(p+2)(p+3)/6$. It is easy to see that they form a basis for $P^p(\Omega_{ref})$, the space of polynomials of total degree p on Ω_{ref} . The union of the closed edges is the *wire basket* of Ω_{ref} .

It turns out that if we use some standard vertex and edge functions [16], the preconditioned system that defines our algorithm is very ill conditioned; cf. [2, Section 5.1.1]. We therefore construct *low energy* vertex and edge functions; see [2, Chapter 4] which result in the bound (1). They satisfy certain stability properties related to their values on the edges and boundary of the reference tetrahedron. Their construction is based on the extension theorems in [1], [12], [10]. To avoid technical details here, we only remark that the low energy functions with highly oscillatory traces on the wire basket decay much more rapidly away from the edges than the standard ones that have the same trace. See Fig. 1 for a comparison of low energy and standard (high energy) basis functions.

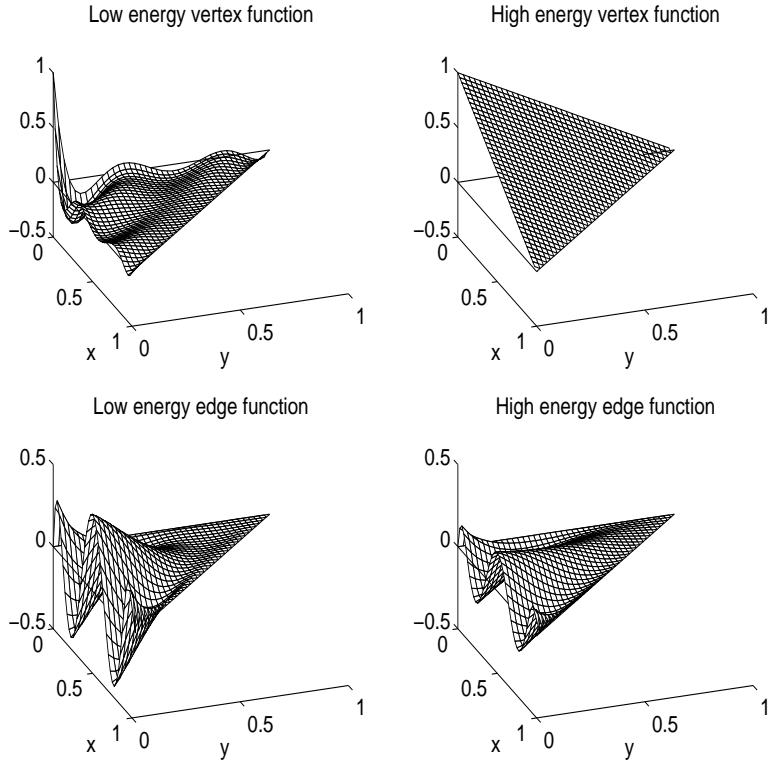


FIGURE 1. Basis functions on the face F_3 of the reference tetrahedron Ω_{ref}

We next eliminate the interior degrees of freedom and end up with a system $Sx_\Gamma = b_\Gamma$, where S is built from local Schur complements, also by subassembly.

3. A wire basket algorithm

This algorithm is similar to the wire basket algorithm defined in [6, Section 6.2] and [13, 14, Section 6]. An interesting theoretical feature of this algorithm is that the bound on the condition number of the global preconditioned system is the same as the local one.

We use a preconditioned conjugate gradient algorithm, with the preconditioner built from blocks that correspond to subspaces. In its simplest form, this algorithm is a block-Jacobi method. We define it in the variational framework known as the abstract Schwarz theory; see, e.g., Smith, Bjørstad, and Gropp [15].

The coarse space V_W is the space spanned by the vertex and edge functions and contains the constants. The construction of such a space is quite intricate; cf [6, Section 6.2], [13, 14, Section 6], [2, Section 4.3].

All the local spaces are associated with individual faces of the tetrahedra. For each face F_k , we define the face space V_{F_k} as the space of functions in V^p , that vanish on all the faces of the interface Γ except F_k . We obviously have

$$V^p = V_W + \sum_k V_{F_k}.$$

TABLE 1. Local condition numbers, wire basket algorithm, low energy vertex and edge functions

p	Constants not in the wire basket			Constants in the wire basket		
	λ_{min}	λ_{max}	κ	λ_{min}	λ_{max}	κ
4	0.1921	1.8000	9.3691	0.1331	2.2549	16.9416
5	0.1358	1.7788	13.1022	0.1063	2.3890	22.4775
6	0.1033	1.8203	17.6186	0.0842	2.4503	29.1136
7	0.0864	1.8205	21.0818	0.0753	2.4996	33.2026
8	0.0740	1.8407	24.8854	0.0655	2.5374	38.7335
9	0.0656	1.8476	28.1508	0.0601	2.5668	42.6989
10	0.0590	1.8588	31.4892	0.0541	2.5911	47.9346

To each subspace $V = V_W$ or V_{F_k} , we associate an operator T_V defined by

$$\hat{a}_V(T_V u, v) = a(u, v) \quad \forall v \in V.$$

Here, $\hat{a}_V(\cdot, \cdot)$ is a positive definite, symmetric bilinear form on the subspace V .

Each bilinear form $\hat{a}_V(\cdot, \cdot)$ uniquely defines the operator T_V and vice-versa. We say that T_V is an approximate projection. If $\hat{a}_V(\cdot, \cdot) = a(\cdot, \cdot)$ then $T_V = P_V$, the $a(\cdot, \cdot)$ -orthogonal projection on V . For specific subspaces, we choose T_V to be almost spectrally equivalent to P_V , but cheaper to compute.

On the coarse space, we can use the exact solver $a(\cdot, \cdot)$ or, more economically, an inexact solver based on the bilinear form

$$(3) \quad \hat{a}_W(u, u) = (1 + \log p) \sum_i \inf_{c_i} \|u - c_i\|_{L^2(W_i)}^2.$$

On each face space, we choose $\hat{a}_{F_k}(\cdot, \cdot) = a(\cdot, \cdot)$.

The *additive Schwarz method* (ASM) is defined by the operator

$$(4) \quad T_a = T_W + T_{F_1} + \dots + T_{F_{nf}},$$

where nf is number of faces in Γ . The equation $T_a u = g_a$, where $g_a = T_W u + T_{F_1} u + \dots + T_{F_{nf}} u$ can be solved by the conjugate gradient method, which can be viewed as a preconditioned conjugate gradient method for the initial system $Sx_\Gamma = b_\Gamma$.

The preconditioner for the additive Schwarz method, using the exact solver on V_W , has the following matrix form:

$$(5) \quad S_{prec} = \begin{pmatrix} S_{WW} & 0 & 0 & 0 \\ 0 & S_{F_1 F_1} & 0 & 0 \\ 0 & 0 & S_{F_2 F_2} & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}.$$

If we use the inexact solver $\hat{a}_W(\cdot, \cdot)$, the block S_{WW} is replaced by

$$\hat{S}_{WW} = (1 + \log p) \left(M - \sum_i \frac{(M^{(i)} z^{(i)}) \cdot (M^{(i)} z^{(i)})^T}{z^{(i)T} M^{(i)} z^{(i)}} \right),$$

where $M^{(i)}$ is the mass matrix of the wire basket W_i , and $z^{(i)}$ is the vector containing the coefficients of the constant function 1. The mass matrix M , for our particular choice of vertex and edge functions, is tridiagonal.

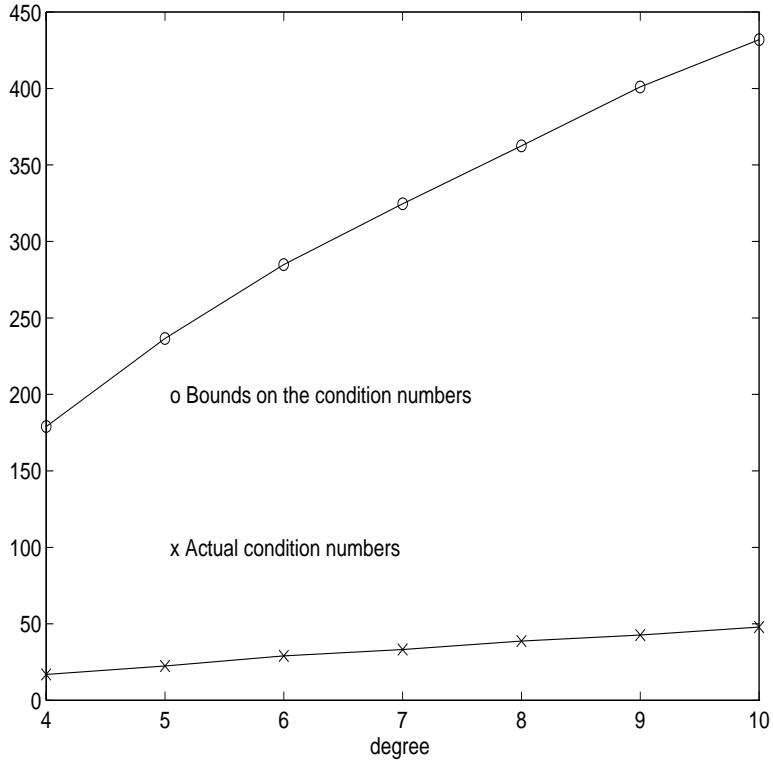


FIGURE 2. Condition numbers and bounds on them, for the wire basket algorithm

The *symmetrized multiplicative method* (MSM) is defined by the operator

$$T_m = I - (I - T_W)(I - T_{F_1}) \cdots (I - T_{F_{nf}}) \cdots (I - T_{F_1})(I - T_W).$$

The *hybrid method* (HSM) is defined by the operator

$$T_h = I - (I - T_W)(I - T_{F_1} - \cdots T_{F_{nf}})(I - T_W).$$

The matrix form of the preconditioners defined by the operators T_m and T_h is not block-diagonal.

4. Numerical experiments

We start by computing the local condition number of the preconditioned Schur complement, on a reference tetrahedron; see Table 1. We obtain lower condition numbers in case when the constants are not in the coarse space space. However, as we mentioned in Section 3, we must add the constants to the coarse space. We do this at the expense of increasing the condition numbers by a factor of 1.5 – 1.6. Next, we look at the bounds on these condition numbers, as given by the theory. To this end, we compute all the constants in the inequalities used in the proof (1); see [2, Section 5.1.2]. The asymptotic logarithmic growth of these bounds is more visible than that of the actual condition numbers; see Fig. 2.

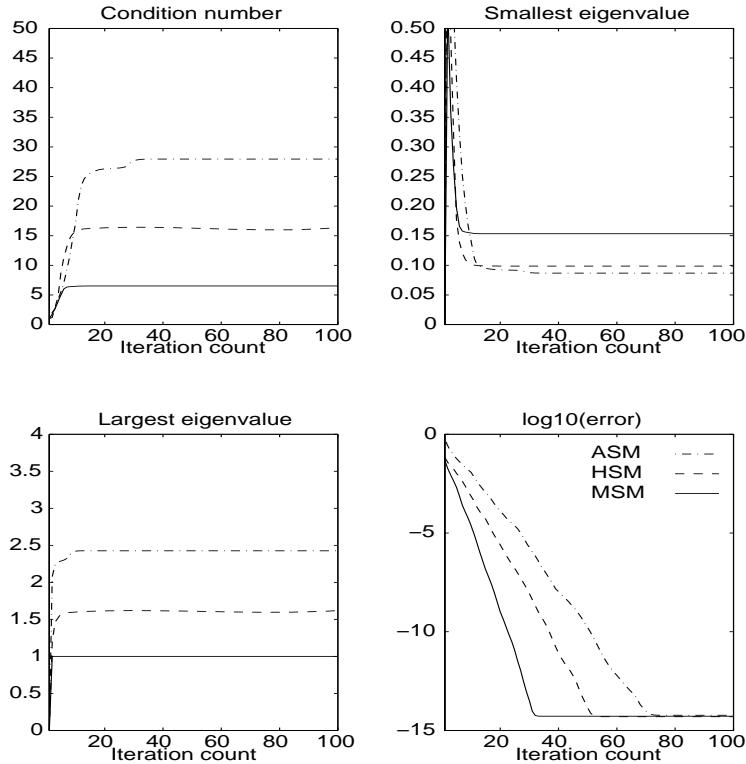


FIGURE 3. Comparison of additive, hybrid, and multiplicative methods, $p = 6$, exact solver on the wire basket.

We now move to global experiments. The number of iterations is fixed beforehand. We compare the performances of the additive, hybrid, and multiplicative methods, see Fig. 3. The extreme eigenvalues are computed via the Lanczos iteration. We have performed experiments on a cubic region that consists of 192 identical tetrahedra, for $p = 6$, with Dirichlet boundary conditions on one face of Ω , and Neumann on the others. We have used the exact solver on the wire basket. We remark that the global condition number of the additive method coincides with the local one, given in Table 1, at the intersection of the last column and the row that corresponds to $p = 6$. We remark that the multiplicative method performs better than the hybrid method, which performs better than the additive one. We can use an inexact solver on the wire basket, which makes the coarse problem cheaper to solve, at some expense in the performance of the full algorithm; see [2, Section 5.2.3].

References

1. Ivo Babuška, Alan Craig, Jan Mandel, and Juhani Pitkäranta, *Efficient preconditioning for the p -version finite element method in two dimensions*, SIAM J. Numer. Anal. **28** (1991), no. 3, 624–661.
2. Ion Bica, *Iterative substructuring methods for the p -version finite element method for elliptic problems*, Ph.D. thesis, New York University, September 1997, Tech. Rep. 1997-743.,

3. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), no. 175, 103–134.
4. ———, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp. **53** (1989), 1–24.
5. Mario A. Casarin, *Diagonal edge preconditioners in p -version and spectral element methods*, Tech. Report 704, Department of Computer Science, Courant Institute, September 1995, To appear in SIAM J. Sci. Comp.
6. Maksymilian Dryja, Barry F. Smith, and Olof B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. **31** (1994), no. 6, 1662–1694.
7. Benqi Guo and Weiming Cao, *A preconditioner for the h - p version of the finite element method in two dimensions*, Numer. Math. **75** (1996), 59–77.
8. ———, *An additive Schwarz method for the hp -version finite element method in three dimensions*, SIAM J. Sci. Comp. (To appear.).
9. ———, *Preconditioning for the h - p version of the finite element method in two dimensions*, SIAM J. Numer. Analysis. (To appear).
10. Yvon Maday, *Relèvement de traces polynomiales et interpolations Hilbertiennes entre espaces de polynômes*, C. R. Acad. Sci. Paris **309**, Série I (1989), 463–468.
11. Jan Mandel, *Two-level domain decomposition preconditioning for the p -version finite element version in three dimensions*, Int. J. Numer. Meth. Eng. **29** (1990), 1095–1108.
12. Rafael Munoz-Sola, *Polynomial liftings on a tetrahedron and applications to the h - p version of the finite element method in three dimensions*, SIAM J. Numer. Anal. **34** (1997), no. 1, 282–314.
13. Luca F. Pavarino and Olof B. Widlund, *A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions*, SIAM J. Numer. Anal. **33** (1996), no. 4, 1303–1335.
14. ———, *Iterative substructuring methods for spectral elements: Problems in three dimensions based on numerical quadrature*, Computers Math. Applic. **33** (1997), no. 1/2, 193–209.
15. Barry F. Smith, Petter Bjørstad, and William Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
16. Barna Szabó and Ivo Babuška, *Finite element analysis*, John Wiley & Sons, New York, 1991.

COURANT INSTITUTE, NEW YORK UNIVERSITY

Current address: Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877

E-mail address: bica@ridgefield.sdr.slb.com

A 2-level and Mixed Domain Decomposition Approach for Structural Analysis

David Dureisseix and Pierre Ladevèze

1. Introduction

When using domain decomposition methods without overlapping, one can focus on displacements, such as primal approaches, [11] . . . , or on efforts, such as dual approaches, [6]. Since the LATIN approach used herein allows interfaces to play a major role, both displacements and efforts are the unknowns; it is a “mixed” approach. A general drawback with domain decomposition methods is the decrease in convergence as increases the number of substructures. Using a global mechanism to propagate information among all substructures can eliminate this drawback.

We are proposing herein to take into account the introduction of two scales when decomposing the structure into substructures and interfaces. As a first step, the implemented version is concerned with linear elasticity. The large scale problem is then used to build a global exchange of information and therefore to improve performance. Moreover, comparisons with other decomposition methods, and in particular with several variants of the FETI method, are proposed.

2. Formulation of the problem

The studied structure is seen as the assembly of two mechanical entities: substructures Ω^E , $E \in \mathbf{E}$, and interfaces $L^{EE'}$. Each possess its own variables and equations. The principles of this one-level approach have been described in [9], its feasibility has been shown in [10], and [2] proposes some significant examples.

Since we are dealing herein with linear elasticity, only the final configuration is of interest.

2.1. Substructure behaviour. Each substructure Ω^E is submitted to the action of its environment (neighbouring interfaces): an effort \underline{F}^E and a displacement field \underline{W}^E on its boundary $\partial\Omega^E$. Eventually, \underline{f}_d is a prescribed body force (Figure 1).

For each $E \in \mathbf{E}$, $(\underline{W}^E; \underline{F}^E)$ has to satisfy:

- kinematic equations:

$$(1) \quad \exists \underline{U}^E \in \mathcal{U}^E, \quad \varepsilon^E = \varepsilon(\underline{U}^E) \quad \text{and} \quad \underline{U}_{|\partial\Omega^E}^E = \underline{W}^E$$

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 73C35, 73V05, 65F10.

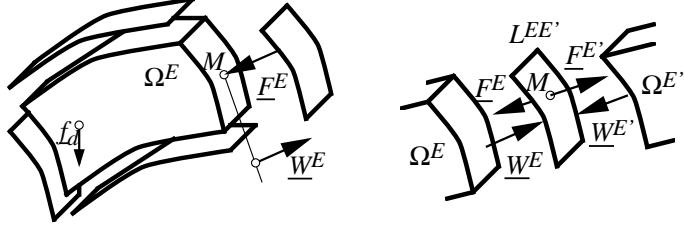


FIGURE 1. Substructure and interface

where \mathcal{U}^E is the set of displacement fields defined on Ω^E which possess a finite energy, and ε^E is the associated strain.

- equilibrium equations: a stress field σ^E balances f_d and F^E , i.e.:

$$(2) \quad \forall \underline{U}^* \in \mathcal{U}^E, \quad \int_{\Omega^E} \text{Tr}[\sigma^E \varepsilon(\underline{U}^*)] d\Omega = \int_{\Omega^E} f_d \cdot \underline{U}^* d\Omega + \int_{\partial\Omega^E} \underline{F}^E \cdot \underline{U}^* dS$$

- constitutive relation: herein, the behaviour is linear and elastic (\mathbf{K} denotes Hooke's tensor) and

$$(3) \quad \sigma^E = \mathbf{K} \varepsilon^E$$

\mathbf{s} denotes the set of unknowns $(\underline{W}^E, \underline{F}^E, \underline{U}^E, \sigma^E)$ for $E \in \mathbf{E}$, that characterises the state of all substructures.

2.2. Interface behaviour. The state of the liaison between two substructures Ω^E and $\Omega^{E'}$ is defined by values on its surface of both the displacements and efforts $(\underline{W}^E; \underline{F}^E)$ and $(\underline{W}^{E'}; \underline{F}^{E'})$ (see Figure 1). For a perfect liaison, they must satisfy:

$$(4) \quad \underline{F}^E + \underline{F}^{E'} = 0 \quad \text{and} \quad \underline{W}^E = \underline{W}^{E'}$$

Of course, other kinds of liaison can be expressed, such as the prescribed effort liaison, the prescribed displacement liaison, and the unilateral contact liaison with or without friction, as described in [9], [2]. Here, we are only dealing with perfect interfaces that continuously transfer both efforts and displacements.

2.3. Description of the one-level algorithm. According to the framework of LArge Time INcrement (LATIN) methods, equations are split into two groups in order to separate difficulties, [10]:

- Γ is the set of unknowns \mathbf{s} satisfying each interface behaviour (4), and
- \mathbf{A}_d is the set satisfying each substructure behaviour (1), (2), (3).

The solution \mathbf{s}_{ex} searched is then the intersection of \mathbf{A}_d and Γ . A two-stage algorithm successively builds an element of \mathbf{A}_d and an element of Γ . Each stage involves a search direction; these are the parameters of the method:

- the local stage uses the search direction \mathbf{E}^+ : $(\hat{\underline{F}} - \underline{F}) - k(\hat{\underline{W}} - \underline{W}) = 0$

Finding $\hat{\mathbf{s}} \in \Gamma$ in such a way that $\hat{\mathbf{s}} - \mathbf{s}_n$ belongs to the search direction \mathbf{E}^+ is a local problem on the interfaces. For instance, with perfect interfaces, the solution is explicitly written: $\hat{\underline{W}} = \hat{\underline{W}'} = \frac{1}{2}[(\underline{W} + \underline{W}') - k^{-1}(\underline{F} + \underline{F}')]$ and $\hat{\underline{F}} = -\hat{\underline{F}'} = \frac{1}{2}[(\underline{F} - \underline{F}') - k(\underline{W} - \underline{W}')]$. It can easily be parallelised.

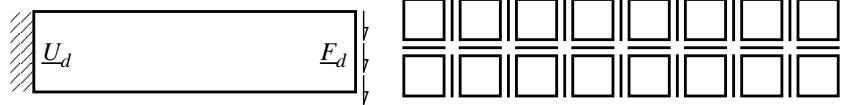


FIGURE 2. Model problem and example of decomposition into 16 substructures and interfaces

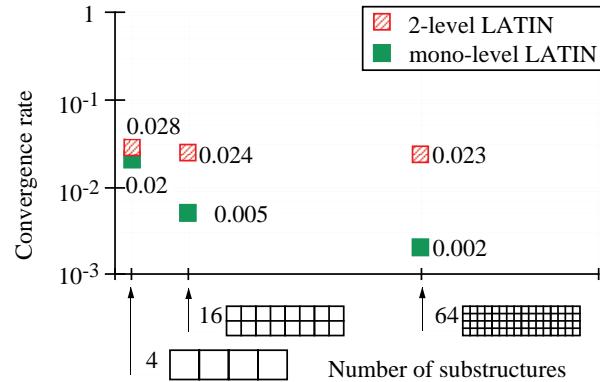


FIGURE 3. Convergence rate versus number of substructures

- the linear stage uses the search direction \mathbf{E}^- : $(\underline{F} - \hat{\underline{F}}) + k(\underline{W} - \hat{\underline{W}}) = 0$. Finding $\mathbf{s}_{n+1} \in \mathbf{A}_d$ in such a way that $\mathbf{s}_{n+1} - \hat{\mathbf{s}}$ belongs to \mathbf{E}^- is a global problem on each substructure. When using the search direction, (2) is an elasticity-like problem on each substructure, with Robin boundary conditions. It can be solved concurrently once the substructures have been distributed among the available processors, along with their neighbouring interfaces.

Finally, a convergence check can be built with $\|\hat{\mathbf{s}} - \mathbf{s}_n\|$. More details for this one-level approach can be found in [10]. In the case of linear elasticity, this algorithm is similar to the one proposed in [8], [12], [7], i.e. it is one version of Uzawa algorithm.

3. A 2-level extension

Let us first consider the model problem of a slendered bidimensional structure submitted to a parabolic bending loading (see Figure 2). The reference $(\underline{U}; \sigma)_{\text{ref}}$ here is the direct finite element solution without decomposition. It allows us to define the convergence rate in energy norm:

$$\tau = -\log \frac{e_{n+1}}{e_n} \quad \text{where} \quad e_n^2 = \frac{\frac{1}{2} \int_{\Omega} \text{Tr}[(\sigma_n - \sigma_{\text{ref}}) \mathbf{K}^{-1} (\sigma_n - \sigma_{\text{ref}})] d\Omega}{\frac{1}{2} \int_{\Omega} \text{Tr}[\sigma_{\text{ref}} \mathbf{K}^{-1} \sigma_{\text{ref}}] d\Omega}$$

Figure 3 presents the averaged convergence rate (up to convergence: $e_n \leq 0.1\%$) versus the number of substructures. It illustrates a well-known behaviour of domain decomposition methods: slowing the convergence rate when increasing the number of subdomains, [1]. To remedy such a drawback, we select herein to express the

solution on two different scales:

$$(5) \quad (\underline{U}^E; \sigma^E) = (\underline{U}_1^E; \sigma_1^E) + (\underline{U}_2^E; \sigma_2^E)$$

1 and 2 denote unknowns related to large scale (effective quantities) and related to corrections on the fine scale respectively. The large scale problem is kept global in order to build the global information exchange mechanism, while the fine scale is managed with the previous substructuring technique.

Each level can arise from a different model for the structure; here, they are related to 2 different meshes with embedded elements. Let Ω_1 and Ω_2 denote these meshes. The principles of such a technique are described in [4]. As in the multigrid terminology, information transfer between levels is performed with a prolongation operator, \mathbf{P} , and a restriction operator, $\mathbf{R} = \mathbf{P}^T$. $(\underline{U}, \bar{\sigma})$ is the effective part of the solution, i.e. the part defined on the mesh Ω_1 (then, $\underline{U}_1^E = \mathbf{P}^E \underline{U}$ and $\bar{\sigma} = \sum_{E \in \mathbf{E}} \mathbf{R}^E \sigma_1^E$). With embedded grids, the prolongation is straightforward and performed with a classical hierarchical finite element projection, as hierarchical bases are used for splitting \mathcal{U}^E into \mathcal{U}_1 and \mathcal{U}_2^E . With such a splitting, the global equilibrium equations become:

$$(6) \quad \forall \underline{U}_1^* \in \mathcal{U}_1, \quad \forall \underline{U}_2^* \in \mathcal{U}_2^E, \quad \sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\sigma(\varepsilon(\underline{U}_1^*) + \varepsilon(\underline{U}_2^*))] d\Omega = \\ = \sum_{E \in \mathbf{E}} \int_{\Omega^E} (\underline{f}_d \cdot \underline{U}_1^* + \underline{f}_d \cdot \underline{U}_2^*) d\Omega + \sum_{E \in \mathbf{E}} \int_{\partial\Omega^E} \underline{F}_2^E \cdot \underline{U}_2^* dS$$

with $\sigma = \sigma_1 + \sigma_2 = \mathbf{K}\varepsilon(\underline{U}_1) + \mathbf{K}\varepsilon(\underline{U}_2)$, and the search direction on fine scale $\underline{F}_2^E = \hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E - k\underline{W}_2^E$, with $\underline{W}_2^E = \underline{U}_2^E|_{\partial\Omega^E}$, it leads to:

- on the fine scale 2, for each substructure Ω_2^E , the stress field σ_2^E also has to balance $-\sigma_1^E = -\mathbf{K}\varepsilon(\underline{U}_1|_{\Omega^E}) = -\mathbf{K}\varepsilon(\mathbf{P}^E \underline{U})$:

$$(7) \quad \forall \underline{U}_2^* \in \mathcal{U}_2^E, \quad \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_2) \mathbf{K}\varepsilon(\underline{U}_2^*)] d\Omega + \int_{\partial\Omega^E} \underline{U}_2 \cdot k\underline{U}_2^* dS = \\ = \int_{\Omega^E} \underline{f}_d \cdot \underline{U}_2^* d\Omega + \int_{\partial\Omega^E} (\hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E) \cdot \underline{U}_2^* dS - \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_1) \mathbf{K}\varepsilon(\underline{U}_2^*)] d\Omega$$

The discretised displacement-oriented formulation of the problem (7) is:

$$(8) \quad ([K^E] + [k^E])[\underline{U}_2^E] = [\underline{f}_d^E] + [\hat{f}^E] - [B_2 \sigma_1^E]$$

$[K^E]$ and $[k^E]$ denote rigidity matrices (constant along iterations), arising from material and search direction respectively, $[\hat{f}^E]$ is a load due to $\hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E$, and B_2 is the operator giving the generalised forces that balance a given stress field on mesh Ω_2^E . We can notice that the problem to solve is global on the substructure and is elasticity-like in nature.

- on the large scale 1,

$$(9) \quad \forall \underline{U}_1^* \in \mathcal{U}_1, \quad \int_{\Omega} \text{Tr}[\varepsilon(\underline{U}_1) \mathbf{K}\varepsilon(\underline{U}_1^*)] d\Omega = \int_{\Omega} \underline{f}_d \cdot \underline{U}_1^* d\Omega - \sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_2) \mathbf{K}\varepsilon(\underline{U}_1^*)] d\Omega$$

TABLE 1. 2-level algorithm

Large scale — 1 processor	Fine scale — n processors
Initialisation initialisation of $[B_1\bar{\sigma}_2] = 0$ receiving $[B_1\bar{\sigma}_d]_{ \Omega^E}$ ← assembling the contributions factorisation of $[K_1]$ forward-backward on (10) sending \bar{U} →	Initialisation computing contributions $[B_1\bar{\sigma}_d]_{ \Omega^E}$ sending $[B_1\bar{\sigma}_d]_{ \Omega^E}$ initialisation of $\hat{s} = 0$ factorisation of $[K^E] + [k^E]$ → receiving \bar{U} computing coupling term $[B_2\sigma_1^E]$ forward-backward on (8)
Loop over iterations receiving $[B_1\bar{\sigma}_2]_{ \Omega^E}$ ← assembling the contributions forward-backward on (10) sending \bar{U} →	Loop over iterations computing coupling term $[B_1\bar{\sigma}_2]_{ \Omega^E}$ sending $[B_1\bar{\sigma}_2]_{ \Omega^E}$ local stage , convergence check ←→ → receiving \bar{U} computing coupling term $[B_2\sigma_1^E]$ forward-backward on (8)

with $\underline{U}_1^\star = \mathbf{P}\bar{U}$, the last term is: $-\sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\mathbf{R}^E \varepsilon(\underline{U}_2) \mathbf{K} \varepsilon(\bar{U}^\star)] d\Omega$. As the stress field $\bar{\sigma}$ must balance $-\bar{\sigma}_2 = -\sum_{E \in \mathbf{E}} \mathbf{R}^E \mathbf{K} \varepsilon(\underline{U}_2^E)$, the scales are not separated. The discretised displacement-oriented formulation of the problem (9), with $\bar{\sigma}_d$ arising from external loads, is:

$$(10) \quad [K_1][\bar{U}] = [B_1\bar{\sigma}_d] - [B_1\bar{\sigma}_2]$$

The solution is searched successively from the two levels within each LATIN iteration on the substructured fine scale, in a fixed point method, as described in [4]. The linear stage is then performed on both scales, while local stage is still the same as for the one-level approach but only deals with fine scale quantities: $(\hat{W}_2^E; \hat{F}_2^E)$. Table 1 describes the algorithm. It has been implemented in the industrial-type code CASTEM 2000 developed at the CEA in Saclay, [13].

For the previous example, the convergence rate has been illustrated in Figure 3. The quasi-independence of the convergence rate with respect to the number of substructures shows the numerical scalability of the 2-level LATIN method. One can notice that for this example, the new optimum value for the search direction is now related to the interface length (L_0 has then be chosen as equal to 0.25 times the length of one substructure). It is no longer characterised by the behaviour of the whole structure [2], but becomes a substructuring characteristic, see [4].

4. Comparisons

Several domain decomposition algorithms currently use a global mechanism, like the FETI method, [6]. It produces at each iteration a solution that satisfies

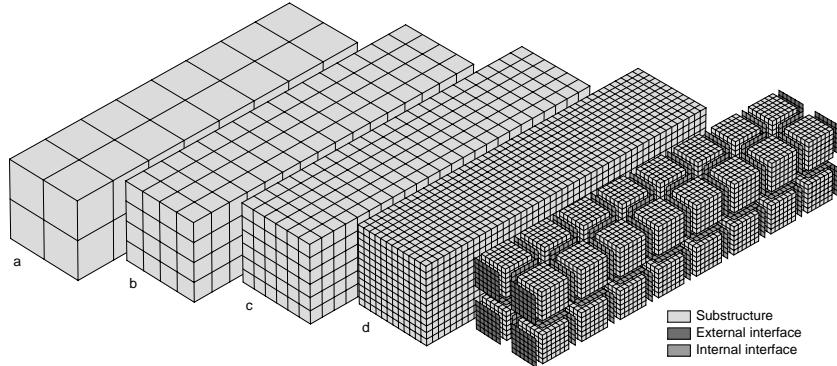


FIGURE 4. Meshes of the large-scale problem (a,b,c,d) and of the substructured problem (assembly of substructures and interfaces)

equilibrium through the interfaces, and that has to accommodate global equilibrium on each subdomain. This leads to the resolution of a global problem on all subdomains to find their rigid body movement, related to the large-scale problem.

The proposed example at this time is a tridimensional beam with a parabolic bending loading at one extremity. 32 substructures and a mesh with 20-node cubic elements are considered for this problem (one substructure has 3 675 d.o.f. and requires 12.8 Mb of storage for the factorised rigidity, while the direct problem has 95 043 d.o.f. and requires 1 252 Mb). For the large scale, the influence of the discretisation with 8-node cubic elements is studied, as also shown in Figure 4.

Figure 5 shows error e_n versus iterations, for the FETI method without preconditioning, then with lumped preconditioning, and finally with optimal Dirichlet preconditioning. These three computations have been performed by F.-X. Roux with the PARAGON machine at ONERA-Châtillon, France. The previous single-level LATIN algorithm as well as the 2-level extension for the different large-scale discretisations are also reported. These computations have been performed on the CRAY-T3D computer at IDRIS in Orsay. Both of these parallel computers have been used with 32 processors. Since time comparisons between two approaches depends on the processor, the intercommunication network, the compilers, disk usage, etc., we retain only the major tendencies by weighting the previous results; after analysing the costly parts of simulations, we identified CPU costs of initialisations for the FETI approach and the LATIN single-level to 1, in terms of CPU equivalent time (accumulated on the 32 processors). Afterwards, the FETI iteration and the 2-level LATIN iteration for the case (a) are identified in terms of cost. Figure 6 then shows the evolution of error versus this CPU equivalent time.

The cost for a direct finite element approach is 18 in terms of CPU equivalent time. When using the multi-frontal scheme, [3], [5], the condensed Schur complement problem has 19 875 d.o.f. and requires 329 Mb of storage. The costs are 3 for local condensations and forward-backward substitutions (which can be performed concurrently) and 2.6 for the resolution of the condensed problem (sequentially). Total cost of the analysis is then 5.6 in CPU equivalent time. The cost of a local condensation is higher than a simple factorisation due to the higher fill-in of the local rigidity matrix (in order to treat the boundary d.o.f. at the end).

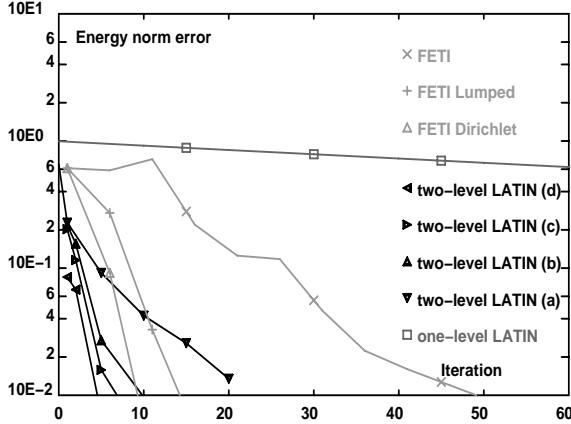


FIGURE 5. Comparison of methods – error versus iterations

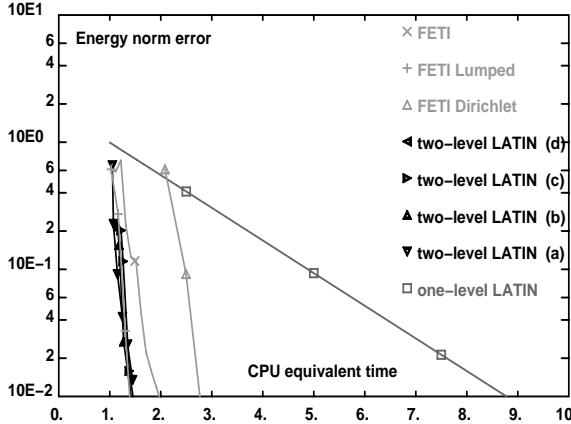


FIGURE 6. Comparison of costs

One can initially note that when increasing the large-scale problem size of the 2-level LATIN algorithm, the error indicator starts out lower at the first iteration because the large-scale first solution is used to initiate the algorithm. Another effect is the increase in the convergence rate (Figure 5), but since iteration costs are also increasing, the two effects cancel each other for the proposed example, (Figure 6).

5. Conclusions

The originality in both the use of the large time increment method and a substructuring approach is the major role played by interfaces, which are considered as structures in their own right. This leads to a “pure parallel” algorithm that can be improved when using a 2-level scheme. The consequence is the generation of a global problem to solve on the whole structure at each iteration. The resulting algorithm is then numerically scalable.

The ultimate goal is the extension to non-linear structural analysis with a large number of d.o.f. One approach which is currently under development deals with a

2-level version more suited to homogenisation techniques, completely merged with non-incremental LATIN methods.

Acknowledgements. The authors wish to thank F.-X. Roux from ONERA-Châtillon, for having performed the computations with FETI approaches on the PARAGON, as well as IDRIS at Orsay, for accessing the CRAY-T3D.

References

1. J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), no. 175, 103–134.
2. L. Champaney, J.-Y. Cognard, D. Dureisseix, and P. Ladevèze, *Large scale applications on parallel computers of a mixed domain decomposition method*, Computational Mechanics (1997), no. 19, 253–263.
3. I. S. Duff, *Parallel implementation of multifrontal schemes*, Parallel Computing **3** (1986), 192–204.
4. D. Dureisseix and P. Ladevèze, *Parallel and multi-level strategies for structural analysis*, Proceedings of the Second European Conference on Numerical Methods in Engineering (J.-A. Désidéri, ed.), Wiley, September 1996, pp. 599–604.
5. Y. Escaig, G. Touzot, and M. Vayssade, *Parallelization of a multilevel domain decomposition method*, Computing Systems in Engineering **5** (1994), no. 3, 253–263.
6. C. Farhat and F.-X. Roux, *Implicit parallel processing in structural mechanics*, Computational Mechanics Advances (J. Tinsley Oden, ed.), vol. 2, North-Holland, June 1994.
7. R. Glowinski and P. Le Tallec, *Augmented Lagrangian interpretation of the nonoverlapping Schwarz alternating method*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1990, pp. 224–231.
8. J. Ladevèze, *Algorithmes adaptés aux calculs vectoriel et parallèle pour des méthodes de décomposition de domaines*, Actes du 3ème colloque Tendances Actuelles en Calcul de Structures (Bastia) (J. P. Grellier and G. M. Campel, eds.), Pluralis, November 1985, pp. 893–907.
9. P. Ladevèze, *Mécanique non-linéaire des structures — nouvelle approche et méthodes de calcul non incrémentales*, Hermès, Paris, 1996.
10. P. Ladevèze and Ph. Lorong, *A large time increment approach with domain decomposition technique for mechanical non linear problems*, Comput. Meths. Appl. Sc. Engng. (New York) (R. Glowinski, ed.), INRIA, Nova Science, 1992, pp. 569–578.
11. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances, vol. 1, North-Holland, 1994.
12. P.-L. Lions, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1990, pp. 202–223.
13. P. Verpeaux, T. Charras, and A. Millard, *CASTEM 2000 : une approche moderne du calcul des structures*, Calcul des Structures et Intelligence Artificielle (J.-M. Fouet, P. Ladevèze, and R. Ohayon, eds.), vol. 2, Pluralis, 1988, pp. 261–271.

LMT CACHAN (E.N.S. DE CACHAN / UNIVERSITÉ PARIS 6 / C.N.R.S.), 61 AVENUE DU PRÉSIDENT WILSON, F-94235 CACHAN CEDEX, FRANCE

E-mail address: dureisse@lmt.ens-cachan.fr

LMT CACHAN (E.N.S. DE CACHAN / UNIVERSITÉ PARIS 6 / C.N.R.S.), 61 AVENUE DU PRÉSIDENT WILSON, F-94235 CACHAN CEDEX, FRANCE

E-mail address: ladeveze@lmt.ens-cachan.fr

Iso-P2 P1/P1/P1 Domain-Decomposition/Finite-Element Method for the Navier-Stokes Equations

Shoichi Fujima

1. Introduction

In recent years, parallel computers have changed techniques to solve problems in various kinds of fields. In parallel computers of distributed memory type, data can be shared by communication procedures called message-passing, whose speed is slower than that of computations in a processor. From a practical point of view, it is important to reduce the amount of message-passing. Domain-decomposition is an efficient technique to parallelize partial differential equation solvers on such parallel computers.

In one type of the domain decomposition method, a Lagrange multiplier for the weak continuity between subdomains is used. This type has the potential to decrease the amount of message-passing since (i) independency of computations in each subdomain is high and (ii) two subdomains which share only one nodal point do not need to execute message-passing each other. For the Navier-Stokes equations, domain decomposition methods using Lagrange multipliers have been proposed. Achdou et al. [1, 2] has applied the mortar element method to the Navier-Stokes equations of stream function-vorticity formulation. Glowinski et al. [7] has shown the fictitious domain method in which they use the constant element for the Lagrange multiplier. Suzuki [9] has shown a method using the iso-P2 P1 element. But the choice of the basis functions for the Lagrange multipliers has not been well compared in one domain decomposition algorithm.

In this paper we propose a domain-decomposition/finite-element method for the Navier-Stokes equations of the velocity-pressure formulation. In the method, subdomain-wise finite element spaces by the iso-P2 P1/P1 elements [3] are used for the velocity and the pressure, respectively. For the upwinding, the upwind finite element approximation based on the choice of up- and downwind points [10] is used. For the discretization of the Lagrange multiplier, three cases are compared numerically. As a result, iso-P2 P1/P1/P1 element shows the best accuracy in a test problem. Speed up is attained with the parallelization.

1991 *Mathematics Subject Classification*. Primary 65M60; Secondary 76D05.

The author was supported by the Ministry of Education, Science and Culture of Japan under Grant-in-Aid for Encouragement of Young Scientists, No.08740146 and No.09740148.

2. Domain decomposition/finite-element method for the Navier-Stokes equations

Let Ω be a bounded domain in R^2 . Let $\Gamma_D (\neq \emptyset)$ and Γ_N be two parts of the boundary $\partial\Omega$. We consider the incompressible Navier-Stokes equations,

$$\begin{aligned} (1) \quad & \partial u / \partial t + (u \cdot \text{grad})u + \text{grad}p = (1/Re)\nabla^2 u + f \quad \text{in } \Omega, \\ (2) \quad & \text{div}u = 0 \quad \text{in } \Omega, \\ (3) \quad & u = g_D \quad \text{on } \Gamma_D, \\ (4) \quad & \sigma \cdot n = g_N \quad \text{on } \Gamma_N, \end{aligned}$$

where u is the velocity, p is the pressure, Re is the Reynolds number, f is the external force, g_D and g_N are given boundary data, σ is the stress tensor and n is the unit outward normal to Γ_N .

We decompose a domain into K non-overlapping subdomains,

$$(5) \quad \overline{\Omega} = \overline{\Omega_1} \cup \dots \cup \overline{\Omega_K}, \quad \Omega_k \cap \Omega_l = \emptyset \quad (k \neq l).$$

We denote by n_k the unit outward normal on $\partial\Omega_k$. If $\overline{\Omega_k} \cap \overline{\Omega_l}$ ($k \neq l$) includes an edge of an element, we say an interface of the subdomains appears. We denote all interfaces by $\Gamma_m, m = 1, \dots, M$. We assume they are straight segments. Let us define integers $\kappa_-(m)$ and $\kappa_+(m)$ by

$$(6) \quad \Gamma_m = \overline{\Omega_{\kappa_-(m)}} \cap \overline{\Omega_{\kappa_+(m)}} \quad (\kappa_-(m) < \kappa_+(m)).$$

Let $\mathcal{T}_{k,h}$ be a triangular subdivision of Ω_k . We further divide each triangle into four congruent triangles, and generate a finer triangular subdivision $\mathcal{T}_{k,h/2}$. We assume that the positions of the nodal points in $\Omega_{\kappa_+(m)}$ and ones in $\Omega_{\kappa_-(m)}$ coincide on Γ_m . We use iso-P2 P1/P1 finite elements [3] for the velocity and the pressure subdomain-wise by

$$\begin{aligned} (7) \quad V_{k,h} &= \{v \in C(\overline{\Omega_k})^2; v|_e \in (P^1(e))^2, e \in \mathcal{T}_{k,h/2}, v = 0 \text{ on } \partial\Omega_k \cap \Gamma_D\}, \\ (8) \quad Q_{k,h} &= \{q \in C(\overline{\Omega_k}); q|_e \in P^1(e), e \in \mathcal{T}_{k,h}\}, \end{aligned}$$

respectively, we construct the finite element spaces by $V_h = \prod_{k=1}^K V_{k,h}$ and $Q_h = \prod_{k=1}^K Q_{k,h}$.

Concerning weak continuity of the velocity between subdomains, we employ the Lagrange multiplier on the interfaces. For the discretization of the spaces of the Lagrange multiplier defined on Γ_m ($1 \leq m \leq M$), we compare three cases (see Figure 1):

Case 1: The conventional iso-P2 P1 element, that is defined by

$$(9) \quad W_{m,h} = (X_{\kappa_+(m),h}|_{\Gamma_m})^2,$$

where $X_{k,h} = \{v \in C(\overline{\Omega_k}); v|_e \in P^1(e), e \in \mathcal{T}_{k,h/2}\}$.

Case 2: A modified iso-P2 P1 element having no freedoms at both edges of interfaces [4].

Case 3: The conventional P1 element, that is defined by

$$(10) \quad W_{m,h} = (Y_{\kappa_+(m),h}|_{\Gamma_m})^2,$$

where $Y_{k,h} = \{v \in C(\overline{\Omega_k}); v|_e \in P^1(e), e \in \mathcal{T}_{k,h}\}$.

The finite element space W_h is defined by $W_h = \prod_{m=1}^M W_{m,h}$.

We consider time-discretized finite element equations derived from (1)-(4):

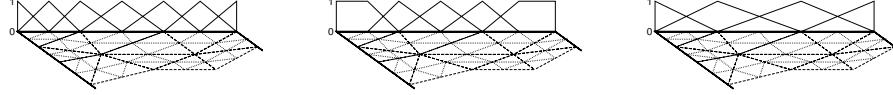


FIGURE 1. Shapes of iso-P2(left), modified iso-P2(center) and P1(right) basis functions for the Lagrange multiplier and a subdivision $\mathcal{T}_{k,h/2}$

PROBLEM 1. Find $(u_h^{n+1}, p_h^n, \lambda_h^n) \in V_h \times Q_h \times W_h$ such that

$$\begin{aligned} \forall v_h \in V_h, \quad (\frac{u_h^{n+1} - u_h^n}{\Delta t}, v_h)_h + b(v_h, p_h^n) + j(v_h, \lambda_h^n) &= \langle \hat{f}, v_h \rangle \\ &\quad - a_1^h(u_h^n, u_h^n, v_h) \\ (11) \quad &\quad - a_0(u_h^n, v_h), \end{aligned}$$

$$(12) \quad \forall q_h \in Q_h, \quad b(u_h^{n+1}, q_h) = 0,$$

$$(13) \quad \forall \mu_h \in W_h, \quad j(u_h^{n+1}, \mu_h) = 0.$$

Forms in Problem 1 are defined by,

$$(14) \quad (u, v) = \sum_{k=1}^K \int_{\Omega_k} u_k \cdot v_k dx,$$

$$(15) \quad a_1(w, u, v) = \sum_{k=1}^K \int_{\Omega_k} (w_k \cdot \text{grad } u_k) v_k dx,$$

$$(16) \quad a_0(u, v) = \frac{2}{Re} \sum_{k=1}^K \int_{\Omega_k} D(u_k) \otimes D(v_k) dx,$$

$$(17) \quad b(v, q) = - \sum_{k=1}^K \int_{\Omega_k} q_k \text{div } v_k dx,$$

$$(18) \quad j(v, \mu) = - \sum_{m=1}^M \int_{\Gamma_m} (v_{\kappa_+(m)} - v_{\kappa_-(m)}) \mu_m ds,$$

$$(19) \quad \langle \hat{f}, v \rangle = \sum_{k=1}^K \left(\int_{\Omega_k} f \cdot v_k dx + \int_{\partial\Omega_k \cap \Gamma_N} g_N \cdot v_k ds \right),$$

$(,)_h$ denotes the mass-lumping corresponding to $(,)$, a_1^h is the upwind finite element approximation based on the choice of up- and downwind points [10] to a_1 , and D is the strain rate tensor.

We rewrite Problem 1 by a matrix form as,

$$(20) \quad \begin{pmatrix} \bar{M} & B^T & J^T \\ B & O & O \\ J & O & O \end{pmatrix} \begin{pmatrix} U^{n+1} \\ P^n \\ \Lambda^n \end{pmatrix} = \begin{pmatrix} F^n \\ 0 \\ 0 \end{pmatrix},$$

where \bar{M} is the lumped-mass matrix, B is the divergence matrix, J is the jump matrix, F^n is a known vector, and U^{n+1}, P^n and Λ^n are unknown vectors. Eliminating U^{n+1} from (20), we get the consistent discretized pressure Poisson equation [8] of a

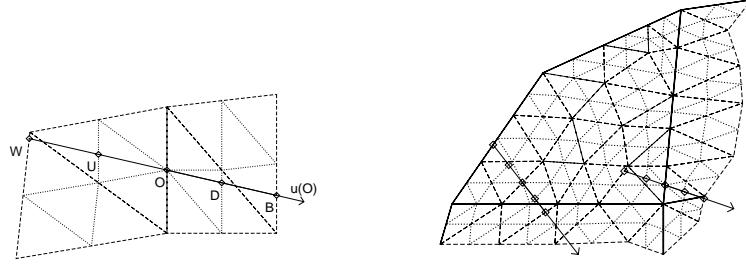


FIGURE 2. Two upwind points(W,U) and two downwind points(D,B) in the finite element approximation based on the choice of up- and downwind points(left) and a domain-decomposition situation(right)

domain-decomposition version. Further eliminating P^n , we obtain a system of linear equations with respect to Λ^n . Applying CG method to this equation, a domain decomposition algorithm [6] is obtained. It is written as follows.

1. $\Lambda^{(0)}$: initial data;
2. Solve $\begin{pmatrix} \bar{M} & B^T \\ B & O \end{pmatrix} \begin{pmatrix} U^{(0)} \\ P^{(0)} \end{pmatrix} = \begin{pmatrix} F - J^T \Lambda^{(0)} \\ O \end{pmatrix}$;
3. $R^{(0)} := -JU^{(0)}$; $\Delta\Lambda^{(0)} := R^{(0)}$; $\rho := (R^{(0)}, \Delta\Lambda^{(0)})$;
4. For $l := 0, 1, 2, \dots$, until $\rho < \varepsilon_{CG}$ do
 - (a) Solve $\begin{pmatrix} \bar{M} & B^T \\ B & O \end{pmatrix} \begin{pmatrix} \Delta U^{(l)} \\ \Delta P^{(l)} \end{pmatrix} = \begin{pmatrix} -J^T \Delta\Lambda^{(l)} \\ O \end{pmatrix}$;
 - (b) $Q := J\Delta U^{(l)}$; $\alpha^{(l)} := \rho / (\Delta\Lambda^{(l)}, Q)$;
 - (c) $(U, P, \Lambda)^{(l+1)} := (U, P, \Lambda)^{(l)} + \alpha^{(l)}(\Delta U, \Delta P, \Delta\Lambda)^{(l)}$;
 - (d) $R^{(l+1)} := R^{(l)} - \alpha^{(l)}Q$;
 - (e) $\mu := (R^{(l+1)}, R^{(l+1)})$; $\beta^{(l)} := \mu / \rho$; $\rho := \mu$;
 - (f) $\Delta\Lambda^{(l+1)} := R^{(l+1)} + \beta^{(l)}\Delta\Lambda^{(l)}$

In Step 2 and 4a, we solve the pressure($P^{(0)}$ or $\Delta P^{(l)}$) separately by the consistent discretized pressure Poisson equation (its matrix is $B\bar{M}^{-1}B^T$) and afterwards we find the velocity($U^{(0)}$ or $\Delta U^{(l)}$). They are subdomain-wise substitution computations since $B\bar{M}^{-1}B^T$ is a diagonal block matrix and it is initially decomposed subdomain-wise in the Cholesky method for band matrices.

REMARK 1. The quantity $\lambda_{m,h}$ corresponds to $\sigma \cdot n_{\kappa_+(m)}|_{\gamma_m}$.

REMARK 2. In the implementation, an idea of two data types is applied to the Lagrange multipliers and the jump matrix. (Each processor handles quantities with respect to $\partial\Omega_k$. They represent either contributive quantities from $\partial\Omega_k$ to $\bigcup_{m=1}^M \Gamma_m$ or restrictive quantities from $\bigcup_{m=1}^M \Gamma_m$ to $\partial\Omega_k$. The detail is discussed in [5].) The idea simplifies the implementation and reduces the amount of message-passing.

REMARK 3. In order to evaluate $a_1^h(u_h^n, u_h^n, v_h)$, we need to find two upwind points and two downwind points for each nodal point (Figure 2(left)). In the domain-decomposition situation, some of these up- and downwind points for nodal points near interfaces may be included in the neighboring subdomains. In order to treat it, each processor corresponding to a subdomain has geometry information

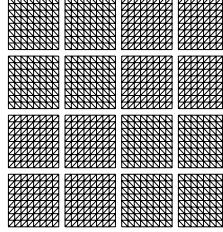


FIGURE 3. An example of domain-decomposition(4×4) and the triangulation($N = 32$)

of all elements which share at least a point with neighboring subdomains (Figure 2(right)). The processors exchange each other the values of u_h^n before the evaluation. Hence the evaluation itself is parallelized without further message-passing.

3. Numerical experiments

3.1. Test problem. Let $\Omega = (0, 1) \times (0, 1)$ and $\Gamma_D = \partial\Omega$ ($\Gamma_N = \emptyset$). The exact stationary solution is $u(x, y) = (x^2y + y^3, -x^3 - xy^2)^T$, $p(x, y) = x^3 + y^3 - 1/2$, and the Reynolds number is set to 400. The boundary condition and the external force are calculated from the stationary Navier-Stokes equations.

We have divided Ω into a union of uniform $N \times N \times 2$ triangular elements, where $N = 4, 8, 16$ or 32 . We have computed in two domain-decomposed ways, where the number of subdomains in each direction is 2 or 4. Figure 3 shows the domain-decomposition and the triangulation in the case $N = 32$ and 4×4 subdomains. Starting from an initial condition for the velocity, the numerical solution is expected to converge to the stationary solution in time-marching. If $\max_{k,i} |u_{k,i}^n - u_{k,i}^{n-1}| / \Delta t < 10^{-5}$ is satisfied, we judge that the numerical solution has converged and stop the computation. Computation parameters are set as $\Delta t = 0.24/N$, $\alpha = 2.0$ and $\varepsilon_{CG} = 10^{-20}$ (α is the stabilizing parameter of the upwind approximation).

Figure 4 shows relative errors between the numerical solutions (u_h, p_h, λ_h) and the exact solution (u, p, λ) . They are defined by

$$|u_h - u|_{V_h} / |u|_{V_h}, \quad \|p_h - p\|_{Q_h} / \|p\|_{Q_h}, \quad \max_m \max_{\Gamma_m} |\lambda_h - \lambda| / \max_{\Omega} p,$$

where

$$|v|_{V_h} = \left\{ \sum_{k=1}^K |v|_{(H^1(\Omega_k))^2}^2 \right\}^{1/2}, \quad \|q\|_{Q_h} = \left\{ \sum_{k=1}^K \|q\|_{L^2(\Omega_k)}^2 \right\}^{1/2}.$$

(We normalize the error of the Lagrange multiplier with $\max_{\Omega} p$, since this quantity is independent of K and the pressure is a dominant term in the stress vector. $|\cdot|_{(H^1(\Omega_k))^2}$ denotes the H^1 semi-norm.) Results of the non-domain-decomposition case are also plotted in the figure. We can observe that the errors of the velocity and the pressure realize the optimal convergence rate of the iso-P2 P1/P1 elements, that is $O(h)$, regardless of choice of $W_{m,h}$. In the first case (iso-P2 P1 element for $W_{m,h}$), the error of the Lagrange multiplier does not converge to 0 when h tends to 0. It may indicate the appearance of some spurious Lagrange multiplier modes, since the degree of freedom of the Lagrange multiplier is larger than that of jump of

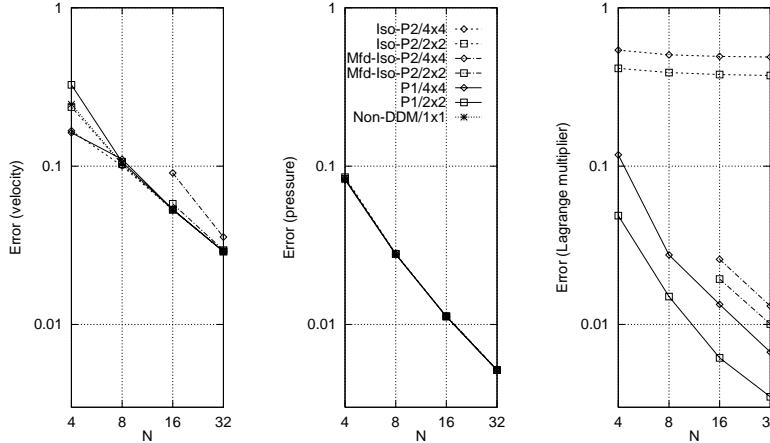


FIGURE 4. Relative errors in the test problem, u_h (left), p_h (center) and λ_h (right)

the velocity in the choice. In the latter two cases the convergence of the Lagrange multiplier has also observed. The third case (P1 element for $W_{m,h}$) shows the best property with respect to the convergence of the Lagrange multiplier.

Since the conventional P1 element has the smallest degree of freedom of the Lagrange multiplier, it can decrease the amount of computation steps in a iteration time in the CG solver. Hence we adopt iso-P2 P1(u)/P1(p)/P1(λ) element in the following.

3.2. Cavity flow problem. We next computed the two-dimensional lid-driven cavity flow problem. The domain $\Omega = (0, 1) \times (0, 1)$ is divided into a uniform $N \times N \times 2$ triangular subdivision, where $N = 24, 48$ or 112 . The Reynolds number is 400 (when $N = 24, 48$) or 1000 ($N = 112$). We chose $\Delta t = 0.01$ ($N = 24$), 0.004 ($N = 48$) or 0.001 ($N = 112$), $\alpha = 2$ and $\varepsilon_{CG} = 10^{-16}$. We computed in several domain-decomposition cases among $1 \times 1, \dots, 8 \times 6, 8 \times 7$ (The case $N = 112$ and 2×2 domain-decomposition was almost full of the memory capacity in the computer we used¹, in this case each subdomain had 6272 elements).

Figure 5(left) shows computation times per a time step (the average of the first 100 time steps). We see that the computation time becomes shorter as the number of subdomains (i.e. processors) increases, except for the non-domain-decomposition case, in which case the performance is almost same with the 2×2 domain-decomposition case. The velocity vectors and the pressure contours of the computed stationary flow in 4×4 subdomains are shown in Figure 6. We can observe that the flow is captured well in the domain decomposition algorithm.

REMARK 4. Since the number of elements in a subdomain is proportional to K^{-1} , the amount of computation per a CG iteration time is in proportion to $K^{-1.5} \sim K^{-1}$ (the former is due to the pressure Poisson equation solver). We have observed that the numbers of CG iteration times per a time step are about $O(K^{0.35})$ when K is large (Figure 5(right)). Thus the amount of computation in a

¹Intel Paragon XP/S in INSAM, Hiroshima University. 56 processors, 16MB memory/proc.

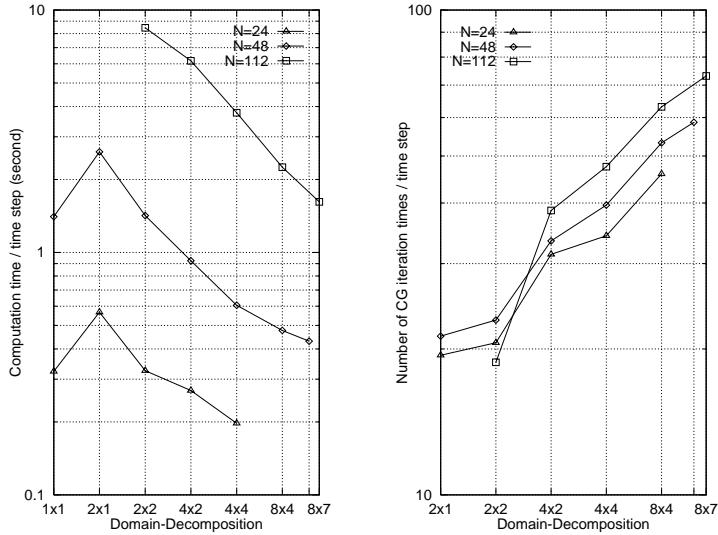


FIGURE 5. Domain-decomposition vs. computation time(left) and the number of CG iteration times(right) per a time step

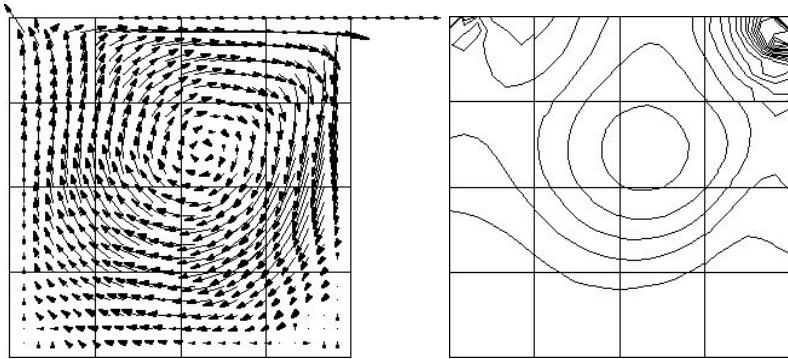


FIGURE 6. Velocity vectors and pressure contour lines of the lid-driven cavity flow problem, $Re = 400$, on a uniform $24 \times 24 \times 2$ triangular subdivision and a 4×4 domain-decomposition

time step is estimated to be proportional to $K^{-1.15} \sim K^{-0.65}$. Obtained speed up, about $O(K^{-0.7})$ in the case of $N = 112$, agrees with the estimation.

4. Conclusion

We have considered a domain decomposition algorithm of the finite element scheme for the Navier-Stokes equations. In the scheme, subdomain-wise finite element spaces by iso-P2 P1/P1 elements are constructed and weak continuity of the velocity between subdomains are treated by a Lagrange multiplier method. This domain decomposition algorithm has advantages such as: (i) each subdomain-wise problem is a consistent discretized pressure Poisson equation so that it is regular,

(ii) the size of a system of linear equations to be solved by the CG method is smaller than that of the original consistent discretized pressure Poisson equation. For the discretization of the Lagrange multiplier, we compared three cases: the conventional iso-P2 P1 element, a modified iso-P2 P1 element having no freedoms at both edges of interfaces, and the conventional P1 element. In every case, we checked numerically in a sample problem that the scheme could produce solutions which converged to the exact solution at the optimal rates for the velocity and the pressure. In the latter two cases we have also observed the convergence of the Lagrange multiplier. Employing the conventional P1 element, we have computed the lid-driven cavity flow problem. The computation time becomes shorter when the number of processor increases.

Acknowledgements

The author wish to thank Professor Masahisa Tabata (Graduate School of Mathematics, Kyushu University) for many valuable discussions and suggestions.

References

1. Y. Achdou and Y. A. Kuznetsov, *Algorithm for a non conforming domain decomposition method*, Tech. Rep. 296, Ecole Polytechnique, 1994.
2. Y. Achdou and O. Pironneau, *A fast solver for Navier-Stokes equations in the laminar regime using mortar finite element and boundary element methods*, SIAM. J. Numer. Anal. **32** (1995), 985–1016.
3. M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the Stokes problem in the primitive variable*, Numer. Math. **33** (1979), 211–224.
4. C. Bernardi, Y. Maday, and A. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, Nonlinear Partial Differential Equations and their Applications (H. Brezis and J. L. Lions, eds.), vol. XI, Longman Scientific & Technical, Essex, UK, 1994, pp. 13–51.
5. S. Fujima, *Implementation of mortar element method for flow problems in the primitive variables*, to appear in Int. J. Comp. Fluid Dyn.
6. ———, *An upwind finite element scheme for the Navier-Stokes equations and its domain decomposition algorithm*, Ph.D. thesis, Hiroshima University, 1997.
7. R. Glowinski, T.-W. Pan, and J. Périoux, *A one shot domain decomposition/fictitious domain method for the Navier-Stokes equations*, Domain Decomposition Methods in Scientific and Engineering Computing, Proc. 7th Int. Conf. on Domain Decomposition (D. E. Keyes and J. Xu, eds.), Contemporary Mathematics, vol. 180, A. M. S., Providence, Rhode Island, 1994, pp. 211–220.
8. P. M. Gresho, S. T. Chan, R. L. Lee, and C. D. Upson, *A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations, part 1: Theory*, Int. J. Num. Meth. Fluids **4** (1984), 557–598.
9. A. Suzuki, *Implementation of domain decomposition methods on parallel computer ADENART*, Parallel Computational Fluid Dynamics: New Algorithms and Applications (N. Satofuka, J. Periaux, and A. Ecer, eds.), Elsevier, 1995, pp. 231–238.
10. M. Tabata and S. Fujima, *An upwind finite element scheme for high-Reynolds-number flows*, Int. J. Num. Meth. Fluids **12** (1991), 305–322.

DEPARTMENT OF MECHANICAL SCIENCE AND ENGINEERING, KYUSHU UNIVERSITY, FUKUOKA
812-8581, JAPAN

Current address: Department of Mathematical Science, Ibaraki University, Mito 310-8512,
Japan

E-mail address: fujima@mito.ipc.ibaraki.ac.jp

Overlapping Nonmatching Grids Method: Some Preliminary Studies

Serge Goossens, Xiao-Chuan Cai, and Dirk Roose

1. Introduction

In this paper, we report some preliminary studies of a finite difference method on overlapping nonmatching grids for a two-dimensional Poisson problem. The method can be regarded as an extension of the Generalised Additive Schwarz Method (GASM). GASM was originally developed as a preconditioning technique that uses special transmission boundary conditions at the subdomain interfaces. By involving a nonmatching grids interpolation operator in the subdomain boundary conditions, we show that the method can also be used as a discretisation scheme. We focus only on the error issues.

2. Generalised Additive Schwarz Method

We first recall briefly the GASM. Suppose we wish to solve $Au = f$ where A represents the discretisation of a PDE defined on a domain which is partitioned into nonoverlapping subdomains. Let $R_i: \Omega \mapsto \Omega_i$ denote the linear restriction operator that maps onto subdomain i by selecting the components corresponding to this subdomain. The matrix $M_i = R_i A R_i^T$ denotes the principal submatrix of the matrix A associated with subdomain Ω_i . The result of applying the GASM can be written as a sum of the solutions of independent subdomain problems, which can be solved in parallel: $M^{-1} = \sum_{i=1}^p R_i^T M_i^{-1} R_i$.

We describe this GASM for the case of two subdomains separated by the interface Γ . A more detailed description has been given by Tan [12] and Goossens et al. [2]. At the heart of the GASM lies an extension of the subdomains to slightly overlapping grids. With a proper definition of the overlap, the restrictions R_i can be defined in such a way that the original discretisation is distributed across the subdomain operators M_i . Figure 1 illustrates the extension process. In case the classical five-point star stencil is used, an overlap of one mesh width is sufficient. After extension towards overlap, and thus duplication of Ω_l and Ω_r into $\Omega_{\bar{l}}$ and $\Omega_{\bar{r}}$ respectively, we obtain an enhanced system of equations $Au = f$ in which we still

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N22, 65F10, 65N06.

Part of this work was carried out during the visit of S. Goossens to the University of Colorado at Boulder. The financial support for this visit by the FWO-Vlaanderen is gratefully acknowledged. This research is also supported by the research Fund of K.U.Leuven (OT/94/16). S. Goossens is financed by a specialisation scholarship of the Flemish Institute for the Promotion of Scientific and Technological Research in Industry (IWT).

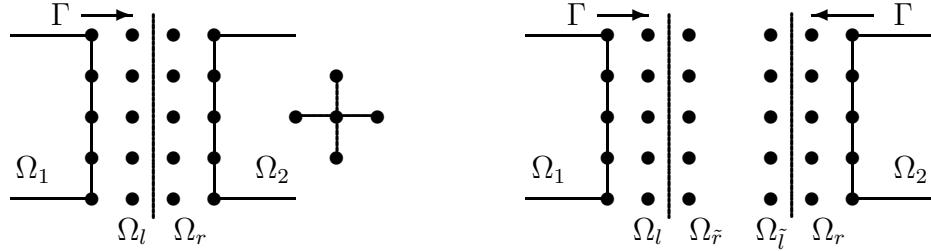


FIGURE 1. Extension of the subdomains to slightly overlapping grids.

have to specify the relation between the overlapping unknowns. The obvious way is just to state that the values in the duplicated subdomains $\Omega_{\tilde{l}}$ and $\Omega_{\tilde{r}}$ should be copied from the values in the original subdomains Ω_l and Ω_r respectively. This is known as the Dirichlet-Dirichlet coupling.

Tang [15] has shown that fast convergence can be obtained by choosing a good splitting, instead of increasing the overlap when a Schwarz enhanced matrix is used. Tan [12] has shown that the spectral properties of the preconditioned operator AM^{-1} and thus the convergence properties of a Krylov subspace method preconditioned by a GASM, are improved by pre-multiplying the enhanced linear system $Au = f$ with a properly chosen nonsingular matrix P . This has been exploited by Goossens et al. [2] to accelerate the solution of the Shallow Water Equations.

This pre-multiplication with P boils down to imposing more general conditions at the subdomain interfaces. This approach has originally been introduced by Lions [5] and subsequently been used by several authors. Hagstrom et. al. [3] advocate the use of nonlocal transmission conditions. Tan and Borsboom [13] have applied the Generalised Schwarz Coupling to advection-dominated problems. Nataf and Rogier [8, 9] have shown that the rate of convergence of the Schwarz algorithm is significantly higher when operators arising from the factorisation of the convection-diffusion operator are used as transmission conditions. Based on these results, Japhet [4] has developed the so-called optimised order 2 (OO2) conditions which result in even faster convergence.

The submatrices C_{lr} , C_{ll} , C_{rr} and C_{rl} represent the discretisation of the transmission conditions and can be chosen freely subject to the condition that the matrix $C = \begin{pmatrix} C_{lr} & -C_{ll} \\ -C_{rr} & C_{rl} \end{pmatrix}$ remains nonsingular. This gives rise to the Generalised Additive Schwarz Preconditioners which are thus based on the enhanced system of equations $Au = f$:

$$(1) \quad \begin{pmatrix} A_{11} & A_{1l} & 0 & 0 & 0 & 0 \\ A_{l1} & A_{ll} & A_{lr} & 0 & 0 & 0 \\ 0 & C_{ll} & C_{lr} & -C_{ll} & -C_{lr} & 0 \\ 0 & -C_{rl} & -C_{rr} & C_{rl} & C_{rr} & 0 \\ 0 & 0 & 0 & A_{rl} & A_{rr} & A_{r2} \\ 0 & 0 & 0 & 0 & A_{2r} & A_{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_l \\ \tilde{u}_r \\ \tilde{u}_l \\ u_r \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_l \\ 0 \\ 0 \\ f_r \\ f_2 \end{pmatrix}.$$

The GASM differs from the classical Additive Schwarz Preconditioner introduced by Dryja and Widlund [1] in that the transmission conditions at the interfaces, i.e. the boundary conditions for the subdomain problems, can be changed in

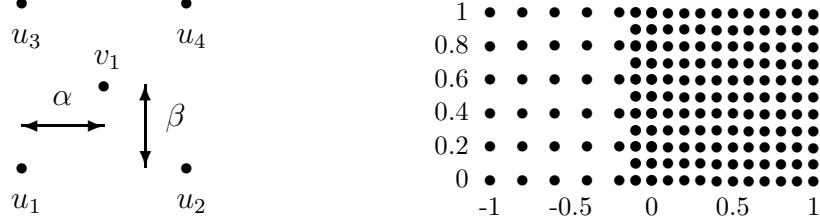


FIGURE 2. Stencil for interpolation and nonmatching grid.

order to improve the spectral properties of the preconditioned operator. An excellent description of the classical Additive Schwarz Preconditioner is given by Smith et al. [10].

3. Nonmatching Grids

The main topic addressed in this paper is a technique aiming at expanding the applicability of this GASM to nonmatching grids. Tan and Borsboom [14] have already shown how to apply the GASM on patched subgrids. The domain they are using, consists of a set of naturally ordered parallelograms, all of which have the same mesh width tangent to the interface. The mesh widths normal to the interface can be different on opposite sides of the interface. We want to alleviate this restriction and present a technique which also allows the GASM to be used when the mesh widths tangent to the interface are different on the opposite sides of the interface. The fact that nonmatching grids are being used implies that interpolation is necessary to transfer information from one grid to the other grid.

3.1. Consistency of grid interpolations. The following definition encapsulates an important concept in the nonmatching grids case.

DEFINITION 1 (Consistent Interpolation). Let $I_{h_j \rightarrow h_i}$ be the interpolation operator from Ω_j to Ω_i with mesh parameters h_j and h_i . Suppose D is the differential operator to be approximated by a finite difference operator $D_i(L_{h_i}, I_{h_j \rightarrow h_i})$, which depends on the usual finite difference operator L_{h_i} and on $I_{h_j \rightarrow h_i}$. We claim that the interpolation operator $I_{h_j \rightarrow h_i}$ is consistent on Ω_i if

$$(2) \quad (D - D_i(L_{h_i}, I_{h_j \rightarrow h_i})) u(x) = O(h_i)$$

for all $x \in \Omega_i$, the part of Ω_i that is overlapped.

The rest of this section is devoted to consistency. Bilinear interpolation is not sufficient for the interpolation operator $I_{h_j \rightarrow h_i}$. Figure 2 shows the stencil. The value of v_1 is given by $v_1 = (1 - \alpha)(1 - \beta)u_1 + \alpha(1 - \beta)u_2 + (1 - \alpha)\beta u_3 + \alpha\beta u_4$ where $\alpha = (x_{v_1} - x_{u_1})/(x_{u_2} - x_{u_1})$ and $\beta = (y_{v_1} - y_{u_1})/(y_{u_3} - y_{u_1})$. If in the discretisation of $-\nabla^2 u$ in Ω_i , the point in Ω_r does not match with a point in Ω_r and its value is computed by bilinear interpolation from points in Ω_r and Ω_2 , then this discretisation is not consistent. Hence higher order interpolation is required.

In [11], a fourth order interpolation formula was constructed with a small interpolation constant for smooth functions satisfying an elliptic equation of the form $-(Au_x)_x - (Bu_y)_y + au = f$, where $A, B > 0$ and $a \geq 0$. This interpolation

formula uses a 4 by 4 stencil. Consequently, the GASM constructed with this interpolation formula requires an extension of at least two grid lines.

Instead of using bilinear interpolation to compute v_1 from u_1, u_2, u_3 and u_4 , which may result in an inconsistent discretisation, we discretise the partial differential equation on the stencil formed by u_1, u_2, u_3, u_4 and v_1 . We seek the coefficients $\gamma_0, \gamma_1, \gamma_2, \gamma_3$ and γ_4 in

$$(3) \quad \begin{aligned} L(\alpha, \beta) = & \gamma_0 u(0, 0) + \gamma_1 u((1-\alpha)h, (1-\beta)h) + \gamma_2 u(-\alpha h, (1-\beta)h) \\ & + \gamma_3 u(-\alpha h, -\beta h) + \gamma_4 u((1-\alpha)h, -\beta h) \end{aligned}$$

so that a consistent approximation to $(u_{xx} + u_{yy})h^2/2$ at $v_1 = u(0, 0)$ is obtained. This can be done using the Taylor expansion for $u(x, y)$ about the origin: $u(x, y) = u + u_{xx}x + u_{yy}y + u_{xx}x^2/2 + u_{xy}xy + u_{yy}y^2/2 + \mathcal{O}(h^3)$. We assume $|x| \leq h$ and $|y| \leq h$ so that the remainder term can be bounded by Ch^3 . The requirements that the coefficients of u, u_x, u_y and u_{xy} vanish together with the requirements that the coefficients of $u_{xx}h^2/2$ and $u_{yy}h^2/2$ equal 1 in the Taylor expansion of (3), yield an overdetermined system $Cg = c$:

$$(4) \quad \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & (1-\beta) & (1-\beta) & -\beta & -\beta \\ 0 & (1-\alpha) & -\alpha & -\alpha & (1-\alpha) \\ 0 & (1-\beta)^2 & (1-\beta)^2 & \beta^2 & \beta^2 \\ 0 & (1-\alpha)^2 & \alpha^2 & \alpha^2 & (1-\alpha)^2 \\ 0 & (1-\alpha)(1-\beta) & -\alpha(1-\beta) & \alpha\beta & -(1-\alpha)\beta \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

This overdetermined system $Cg = c$ can only have solutions if the determinant of the enhanced matrix $\tilde{C} = (C \mid c)$ is zero:

$$(5) \quad \det \tilde{C} = \det(C \mid c) = (\beta - \alpha)(\alpha + \beta - 1).$$

Hence a solution can only exist when $\alpha = \beta$ or $\alpha + \beta = 1$, i.e. when the point v_1 lies on one of the diagonals of the square, formed by u_1, u_2, u_3 and u_4 . The solution is

$$(6) \quad g = (-2 - (1-\alpha)/\alpha - \alpha/(1-\alpha) \quad \alpha/(1-\alpha) \quad 1 \quad (1-\alpha)/\alpha \quad 1)^T$$

when $\alpha = \beta$ and when $\alpha + \beta = 1$ it is

$$(7) \quad g = (-2 - (1-\alpha)/\alpha - \alpha/(1-\alpha) \quad 1 \quad (1-\alpha)/\alpha \quad 1 \quad \alpha/(1-\alpha))^T.$$

The truncation error is determined by substitution of this solution in (3):

$$(8) \quad L(\alpha, \beta) = h^2(u_{xx} + u_{yy})/2 + C_\alpha h^3(u_{xxx} + u_{yyy})/6 + \mathcal{O}(h^4),$$

where $C_\alpha = 1 - 2\alpha$ in case $\alpha = \beta$ and $C_\alpha = 2\alpha - 1$ when $\alpha + \beta = 1$. Hence an $\mathcal{O}(h)$ approximation to $u_{xx} + u_{yy}$ can be obtained, an $\mathcal{O}(h^2)$ approximation can only be obtained when $\alpha = \beta = 1/2$:

$$(9) \quad L(\alpha, \beta) = h^2(u_{xx} + u_{yy})/2 + h^4(u_{xxxx} + 6u_{xxyy} + u_{yyyy})/96 + \mathcal{O}(h^6).$$

In summary, a consistent discretisation exists only if v_1 is in the center or on one of the diagonals of the square formed by u_1, u_2, u_3 and u_4 . The truncation error is $O(h^2)$ when v_1 is in the center and is $O(h)$ when v_1 is on one of the diagonals.

3.2. Error Analysis. Miller [6] has proven the convergence of the Schwarz algorithm based on a maximum principle. We restrict ourselves here to showing that second order accuracy in the L_∞ norm is obtained when a consistent interpolation is used. The convergence of the GASM will be studied elsewhere.

We denote by $p = (i, j)$ an index pair. J_Ω is the set of index pairs of grid points in the domain Ω . We make the following assumptions.

1. For all $p \in J_\Omega$: $\mathcal{L}_h u_p = -c_p u_p + \sum_k c_k u_k$ where the coefficients are positive and the sum over k is taken over mesh points which are neighbours of p .
2. For all $p \in J_\Omega$: $c_p \geq \sum_k c_k$.
3. The set J_Ω is *connected*. By definition a point is connected to each of its neighbours occurring in (1) with a nonzero coefficient. By definition a set is connected if, given any two points p and q in J_Ω , there is a sequence of points $p = p_0, p_1, \dots, p_{m+1} = q$, such that each point p_i is connected to p_{i-1} and p_{i+1} , for $i = 1, 2, \dots, m$.
4. At least one of the equations must involve a Dirichlet boundary condition.

The maximum principle as given by Morton and Meyers [7] can briefly be stated as follows.

LEMMA 2 (Maximum Principle [7]). *Suppose that \mathcal{L}_h , J_Ω and $J_{\partial\Omega}$ satisfy all the assumptions mentioned above and that a mesh function u_p satisfies $\mathcal{L}_h u_p \geq 0$ for all $p \in J_\Omega$. Then u_p cannot attain a nonnegative maximum at an interior point:*

$$(10) \quad \max_{p \in J_\Omega} u_p \leq \max \{ \max_{a \in J_{\partial\Omega}} u_a, 0 \}.$$

THEOREM 3. *Suppose a nonnegative mesh function Φ_p is defined on $J_\Omega \cup J_{\partial\Omega}$ such that $\mathcal{L}_h \Phi_p \geq 1$ for all $p \in J_\Omega$ and that all the assumptions mentioned above are satisfied. Then the error in the approximation is bounded by*

$$(11) \quad |e_p| \leq \max_{a \in J_{\partial\Omega}} \Phi_a \max_{p \in J_\Omega} |T_p|$$

where T_p is the truncation error.

To prove second order accuracy in the L_∞ norm, we show that the discretisation of $\nabla^2(-u) = f$ and the coupling equations satisfy the assumptions (1) and (2) for the maximum principle. The comparison function is chosen as $\Phi(x, y) = ((x - \mu)^2 + (y - \nu)^2) / 4$, resulting in $\mathcal{L}_h \Phi_p = 1$ for all $p \in J_\Omega$. The scalars μ and ν are chosen to minimise the maximum value of this function $\Phi(x, y)$ on the boundary $\partial\Omega$. The classical five-point discretisation of $\nabla^2 u$

$$(12) \quad \mathcal{L}_{h_i} u_p = (u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}) / h_i^2$$

satisfies the assumptions for the maximum principle. In case (9) is used to obtain an equation for v_1 , the assumptions (1) and (2) for the maximum principle are satisfied since this equation has $c_k = 1/(2h_i^2)$ and $c_p = 4/(2h_i^2)$. For problems with at least one Dirichlet boundary condition, the standard error analysis using the maximum principle yields second order accuracy in the L_∞ norm. The proof is essentially the same as the one given by Morton and Meyers [7].

If the point v_1 is not in the center of the square formed by u_1 , u_2 , u_3 and u_4 , we have to use (8) to obtain an equation from which v_1 can be determined. In this case we still have second order accuracy, but a different comparison function must be defined in Ω_l . This is analogous to the classical result that second order accuracy is obtained with a second order discretisation of the partial differential equation and only first order discretisation of the boundary conditions.

TABLE 1. Results for $u_1(x, y) = \exp(-x^2 - y^2)$.

Results for $h_0 = 2h_1$.					
n_0	n_1	L_∞ in block 0	ratio	L_∞ in block 1	ratio
6	11	0.00173386		0.00160317	
11	21	0.000492551	3.52016	0.000439111	3.65094
21	41	0.000128595	3.83025	0.000113528	3.86787
41	81	3.28033e-05	3.92018	2.87286e-05	3.95174
81	161	8.28181e-06	3.96089	7.21997e-06	3.97905
n_0	n_1	L_2 in block 0	ratio	L_2 in block 1	ratio
6	11	0.0006622		0.00051287	
11	21	0.000194007	3.41328	0.000137398	3.73273
21	41	5.20312e-05	3.72867	3.51015e-05	3.91431
41	81	1.34408e-05	3.87114	8.8388e-06	3.9713
81	161	3.41357e-06	3.93746	2.2156e-06	3.98935
Reference results for $h_0 = h_1$.					
n_0	n_1	L_∞ in block 0	L_2 in block 0	L_∞ in block 1	L_2 in block 1
6	6	0.00288425	0.0010893	0.00288425	0.00117384
11	11	0.000736578	0.000280968	0.000736578	0.000299917
21	21	0.000185341	7.12195e-05	0.000185341	7.41716e-05
41	41	4.63929e-05	1.79235e-05	4.63929e-05	1.83303e-05
81	81	1.16036e-05	4.49551e-06	1.16036e-05	4.54876e-06
161	161	2.90103e-06	1.1257e-06	2.90103e-06	1.1325e-06

4. Numerical Examples

The testcases are concerned with the solution of

$$(13) \quad -\nabla^2 u = f \text{ on } \Omega \text{ and } u = g \text{ on } \partial\Omega.$$

The domain $\Omega = \Omega_0 \cup \Omega_1$ consists of two subdomains $\Omega_0 = (-1, 0) \times (0, 1)$ and $\Omega_1 = (-h_1, 1) \times (0, 1)$. The coordinates of the grid points are (x_i, y_j) , where $x_i = x_{\text{ref}} + ih$ and $y_j = y_{\text{ref}} + jh$, for $i = 0, 1, \dots, (n_0 - 1)$ for block 0; $i = 0, 1, \dots, n_1$ for block 1 and $j = 0, 1, \dots, (n - 1)$. The reference point for block 0 is $(-1, 0)$ and for block 1 it is $(-h_1, 0)$. The grid sizes are $h_0 = 1/(n_0 - 1)$ and $h_1 = 1/(n_1 - 1)$. The interface Γ is defined by $x = -h_1/2$. The right-hand side f and the boundary conditions g are chosen such that the exact solution is u_1 resp. u_2 in the testcases, where $u_1(x, y) = \exp(-x^2 - y^2)$ and $u_2(x, y) = \exp(\alpha x) \sin(\beta y)$, where $\alpha = 2$ and $\beta = 8\pi$.

By definition the error is $e_{i,j} = u(x_i, y_j) - u_{i,j}$ where $u(x_i, y_j)$ is the exact solution and $u_{i,j}$ is the computed approximation. In Tables 1–3 we list both the L_∞ norm and L_2 norm of the error, defined by $L_\infty(e) = \max_{i,j} |e_{i,j}|$ and $L_2(e) = \sqrt{\frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} e_{i,j}^2}$.

The results for $-\nabla^2 u_1 = f_1$ on a nonmatching grid are given in Table 1. We also give the results for the same problem on matching grids. This allows us to verify the accuracy of the results. In Table 2 we give the results for $-\nabla^2 u_2 = f_2$. The ratios in the fourth and sixth columns approach 4 as the mesh widths are divided by 2, showing that the method is second order accurate.

To emphasize the importance of consistent interpolation, we give in Table 3 the results for $-\nabla^2 u_2 = f_2$ when bilinear interpolation is used. In this case (2) is not

TABLE 2. Results for $u_2(x, y) = \exp(2x) \sin(8\pi y)$.

Results for $h_0 = 2h_1$.					
n_0	n_1	L_∞ in block 0	ratio	L_∞ in block 1	ratio
6	11	7.23983		4.21601	
11	21	0.48669	14.8756	0.718661	5.86648
21	41	0.102748	4.73673	0.17356	4.14071
41	81	0.0249472	4.11862	0.0453624	3.82608
81	161	0.00656458	3.80027	0.011314	4.0094
n_0	n_1	L_2 in block 0	ratio	L_2 in block 1	ratio
6	11	3.20061		1.66157	
11	21	0.204226	15.6719	0.303296	5.47838
21	41	0.0435961	4.6845	0.0748621	4.0514
41	81	0.01074	4.05923	0.0188233	3.9771
81	161	0.00269816	3.98049	0.00473253	3.97743
Reference results for $h_0 = h_1$.					
n_0	n_1	L_∞ in block 0	L_2 in block 0	L_∞ in block 1	L_2 in block 1
6	6	18.0965	6.69002	47.1923	21.3855
11	11	0.717338	0.264132	3.3735	1.40949
21	21	0.136687	0.0502781	0.718661	0.301632
41	41	0.0321275	0.011811	0.17356	0.074284
81	81	0.00831887	0.00290772	0.0453624	0.0186726
161	161	0.00207194	0.000724103	0.011314	0.00469548

TABLE 3. Results for $u_2(x, y) = \exp(2x) \sin(8\pi y)$ when bilinear interpolation is used.

n_0	n_1	L_∞ in block 0	L_∞ in block 1	L_2 in block 0	L_2 in block 1
7	12	9.63784	7.65477	4.32959	2.7709
12	22	0.847174	0.995454	0.262247	0.366744
22	42	0.142845	0.220307	0.048968	0.082186
42	82	0.0315985	0.0496966	0.011165	0.0197278
82	162	0.00710074	0.0118877	0.00271561	0.00485607

satisfied. The results are for the same problem but solved on shifted grids. The reason for using shifted grids is that an interpolation is required for every point in $\Omega_{\tilde{r}}$, while for grids as in Fig. 2 only half of the points in $\Omega_{\tilde{r}}$ require an interpolation since the other points match some point in Ω_r . The coordinates of the grid points are now given by (x_i, y_j) where $x_i = x_{\text{ref}} + (i - \frac{1}{2})h$ and $y_j = y_{\text{ref}} + (j - \frac{1}{2})h$ for $i = 0, 1, \dots, (n-1)$ and $j = 0, 1, \dots, (n-1)$. The reference point for block 0 is $(-1, 0)$ and for block 1 it is $(0, 0)$. The grid sizes $h_0 = 1/(n_0-2)$ and $h_1 = 1/(n_1-2)$ are the same as in the previous case. Since an inconsistent interpolation is used, the error is larger.

5. Concluding Remarks

We studied an overlapping nonmatching grids finite difference method. A consistency condition is introduced for the nonmatching grids interpolation operator, and under the consistency condition we proved second order global accuracy of the discretisation scheme.

References

1. M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. Report 339, Department of Computer Science, Courant Institute, 1987.
2. S. Goossens, K. Tan, and D. Roose, *An efficient FGMRES solver for the shallow water equations based on domain decomposition*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (P. Bjørstad, M. Espedal, and D. Keyes, eds.), 1996.
3. T. Hagstrom, R. P. Tewarson, and A. Jazcilevich, *Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems*, Appl. Math. Lett. **1** (1988), no. 3, 299–302.
4. C. Japhet, *Optimized Krylov-Ventcell method. Application to convection-diffusion problems*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (P. Bjørstad, M. Espedal, and D. Keyes, eds.), 1996.
5. P. L. Lions, *On the Schwarz alternating method III: A variant for nonoverlapping subdomains*, Proc. Third Int. Conf. on Domain Decomposition Meths. (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1990, pp. 202–223.
6. K. Miller, *Numerical analogs to the Schwarz alternating procedure*, Numer. Math. **7** (1965), 91–103.
7. K. W. Morton and D. F. Mayers, *Numerical solution of partial differential equations*, Cambridge University Press, 1994.
8. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Mathematical Models and Methods in Applied Sciences **5** (1995), no. 1, 67–93.
9. ———, *Outflow boundary conditions and domain decomposition method*, Domain Decomposition Methods in Scientific and Engineering Computing (Providence) (D. E. Keyes and J. Xu, eds.), Contemporary Mathematics, no. 180, AMS, 1995, pp. 289–293.
10. B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
11. G. Starius, *Composite mesh difference methods for elliptic boundary value problems*, Numer. Math. **28** (1977), 243–258.
12. K. H. Tan, *Local coupling in domain decomposition*, Ph.D. thesis, Universiteit Utrecht, 1995.
13. K. H. Tan and M. J. A. Borsboom, *On generalized Schwarz coupling applied to advection-dominated problems*, Domain Decomposition Methods in Scientific and Engineering Computing (Providence) (D. E. Keyes and J. Xu, eds.), Contemporary Mathematics, no. 180, AMS, 1995, pp. 125–130.
14. ———, *Domain decomposition with patched subgrids*, Domain Decomposition Methods in Sciences and Engineering (Chichester) (R. Glowinski, J. Periaux, Z-C. Shi, and O. Widlund, eds.), John Wiley & Sons Ltd., 1997, pp. 117–124.
15. W. P. Tang, *Generalized Schwarz Splittings*, SIAM J. Sci. Stat. Comput. **13** (1992), no. 2, 573–595.

DEPARTMENT OF COMPUTER SCIENCE, KATHOLIEKE UNIVERSITEIT LEUVEN, CELESTIJNENLAAN 200A, B-3001 HEVERLEE, BELGIUM

E-mail address: Serge.Goossens@cs.kuleuven.ac.be

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF COLORADO AT BOULDER, CAMPUS BOX 430, BOULDER, COLORADO 80309-0430, USA

E-mail address: cai@cs.colorado.edu

DEPARTMENT OF COMPUTER SCIENCE, KATHOLIEKE UNIVERSITEIT LEUVEN, CELESTIJNENLAAN 200A, B-3001 HEVERLEE, BELGIUM

E-mail address: Dirk.Roose@cs.kuleuven.ac.be

Nonconforming Grids for the Simulation of Fluid-Structure Interaction

Céline Grandmont and Yvon Maday

1. Introduction

Fluid structure interaction phenomena occur in a large number of applications and the literature on the subject is quite important, both from the practical and implementation point of view. Nevertheless, most of the applications are focused on a particular range of situations in which the domain that is occupied by the fluid is essentially assumed to be independent on time. Recently, a lot of effort has been made on the numerical simulations of fluid structure interactions in the case where this assumption is no more true and, in particular, in situations where the shape of the domain occupied by the fluid is among the unknowns of the problem. We refer for instance to the works [9, 10, 11] and also to some web pages¹ where medical and engineering applications are displayed. We refer also to [6] for an analysis of the mathematical problem.

This new range of applications is made possible thanks to the increase in computing power available and the recent advances in CSD and CFD. Indeed, the current implementations for the simulation of the coupled phenomena are mostly based on the effective coupling of codes devoted to fluid simulations for the ones, and structure simulations for the others. Such a coupling procedure allows for flexibility in the choice of the separate constitutive laws and modelisations of the fluid and structure separately and allows also for the rapid development of the simulation of the interaction phenomenon. This flexibility is however at the price of the definition of correct *decoupling* algorithms of the different codes that lead to a resolution of the coupled situation. In this direction, some attention has to be given for the time *decoupling* and we refer to [11, 10, 8] for numerical analysis of this part. Another problem has to be faced which consists in the coupling of the spatial discretizations. This difficulty, already present in the former works (where the shape of the fluid part is fixed), is certainly enhanced now that the time dependency has increased by one order of magnitude the size of the computations. Indeed, it is mostly impossible to afford the same mesh size on the fluid and on the structure computational domains, especially in three dimensional situations.

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 65M60, 46E25, 20C20.
¹www.crs4.it, www.science.gmu.edu/~rlohner/pages/lohner.html

Another reason for this difficulty appears when different definitions of finite elements are naturally introduced in the structure (hermitian for plates) and the fluid (lagrangian for fluid).

The problem of coupling different discretizations occurs not only in the case of the interaction of different phenomena, but also in cases where, taking benefit of a domain decomposition, one wants to use different discretizations on different subdomains so as to optimise the discretization parameters and the final CPU time. In cases where, a priori, (exact) continuity should be imposed on the unknown solutions we have to face to the same difficulty as before for similar reasons. The mortar element method [1] has been proposed in this frame to produce an optimal approximation in case of variational approximations of elliptic and parabolic partial differential equations.

In this paper we state the main results concerning different ways of imposing these different discrete continuities with a particular interest to the coupling conditions that lead to the optimality of the global approximation.

The modelization that we shall consider here is the two or three dimensional incompressible Navier Stokes equations, for the fluid, and a linear elasticity for the structure. In addition, the structure will be assumed to be of small thickness in one dimension so that a one or two dimensional behaviour of beam or plate type will be used.

2. Formulation of the continuous and discrete problems

Let $\Omega_F(t)$ be the (unknown) domain occupied by the fluid at any time t during the simulation, we consider that the boundary $\partial\Omega_F(t)$ is decomposed into two open parts :

$$\partial\Omega_F(t) = \bar{\Gamma}_0 \cup \bar{\Gamma}(t)$$

where Γ_0 is independent of time and the (unknown) part $\Gamma(t)$ is the interface between the solid and the fluid. Note that $\Gamma(t)$ coincides with the position of the structure at time t . In this fluid domain we want to solve the Navier Stokes equations : Find \mathbf{u} and p such that

$$(1) \quad \begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + \nabla p + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} \quad \text{in } \Omega_F(t), \\ \operatorname{div} \mathbf{u} &= 0 \quad \text{in } \Omega_F(t), \end{cases}$$

these equations are complemented with appropriate boundary conditions over Γ_0 (that we shall take here as being of homogeneous Dirichlet type on the velocity) and of coupling type over $\Gamma(t)$ (that we shall explicit in a while). We consider now the structure part, that is assumed to be set in a Lagrangian formulation, i.e. the unknowns will be the displacement of the structure points with respect to a reference configuration. Under the assumptions we have done on the structure, we have a reference set Ω_S^0 (that can be the position of the structure at rest) and the position of the structure in time is parametrised by the mapping

$$\mathbf{x} \mapsto \mathbf{x} + \mathbf{d}(\mathbf{x}, t)$$

from Ω_S^0 onto $\Omega_S(t)$ that coincides with $\Gamma(t)$. The equations on \mathbf{b} is here implicitly given in an abstract variational framework : find \mathbf{d} such that $\mathbf{d}(\cdot, t) \in Y$ and for

any $\mathbf{b} \in Y$

$$(2) \quad \int_{\Omega_S^0} \frac{\partial^2 \mathbf{d}}{\partial t^2}(\mathbf{x}, t) \mathbf{b}(\mathbf{x}) d\mathbf{x} + a(\mathbf{d}(., t), \mathbf{b}) = G(\mathbf{b})(t)$$

Here Y is some appropriate Hilbert space and G is the outside forcing term that is applied to the structure. We shall assume in what follows that this forcing term only results from the interaction with the fluid and is equal to the fluid stresses on the interface $\Gamma(t)$. The bilinear form a is assumed to take into account the elastic behaviour of the structure and is assumed to be elliptic over Y . The remaining constraint is the coupling between the fluid velocity and the displacement. Actually we want to express the fact that the fluid sticks to the boundary $\Gamma(t)$, and thus

$$\forall \mathbf{x} \in \Omega_S^0, \quad \mathbf{u}(\mathbf{x} + \mathbf{d}(\mathbf{x}, t), t) = \frac{\partial \mathbf{d}}{\partial t}(\mathbf{x}, t)$$

Assuming that we are able to give a proper definition to the space $L^2(\Omega_F(t))$ of all measurable functions defined over $\Omega_F(t)$ with square integrable and $H^1(\Omega_F(t))$ its subspace of all elements the gradient of which belongs to $L^2(\Omega_F(t))$, we first set $H_{0,\Gamma_0}^1(\Omega_F(t))$ as the subspace of $H^1(\Omega_F(t))$ of all functions that vanish over Γ_0 , then $X(t) = (H_{0,\Gamma_0}^1(\Omega_F(t)))^2$, and we propose a global variational formulation of the coupled problem : find $(\mathbf{u}, p, \mathbf{d})$ with

$$(3) \quad \begin{aligned} \forall t, \quad & \mathbf{u}(., t) \in X(t) \\ \forall t, \quad & p(., t) \in L^2(\Omega_F(t)) \\ \forall t, \quad & \mathbf{d}(., t) \in Y \\ \forall t, \quad & \forall \mathbf{x} \in \Omega_S^0, \quad \mathbf{u}(\mathbf{x} + \mathbf{d}(\mathbf{x}, t), t) = \frac{\partial \mathbf{d}}{\partial t}(\mathbf{x}, t) \end{aligned}$$

such that, for any (\mathbf{v}, \mathbf{b}) in the coupled space V defined as

$$V = \{(\mathbf{v}, \mathbf{b}) \in X(t) \times Y / \mathbf{v}(\mathbf{x} + \mathbf{d}(\mathbf{x}, t)) = \mathbf{b}(\mathbf{x}), \forall \mathbf{x} \in \Omega_S^0\},$$

the following equation holds

$$(4) \quad \begin{aligned} \int_{\Omega_F(t)} \frac{\partial \mathbf{u}}{\partial t} \mathbf{v} + \nu \int_{\Omega_F(t)} \nabla \mathbf{u} \nabla \mathbf{v} + \int_{\Omega_F(t)} \mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{v} + \int_{\Omega_F(t)} p \nabla \cdot \mathbf{v} \\ + \int_{\Omega_S^0} \frac{\partial^2 \mathbf{d}}{\partial t^2} \mathbf{b} + a(\mathbf{d}, \mathbf{b}) = \int_{\Omega_F(t)} \mathbf{f} \mathbf{v}, \quad \forall (\mathbf{v}, \mathbf{b}) \in V, \\ \int_{\Omega_F(t)} \nabla \cdot \mathbf{u} q = 0, \quad \forall q \in L^2(\Omega_F(t)), \end{aligned}$$

It has already been noticed (see eg [3] and [6]) that, provided that a solution exists to this system, it is stable in the following sense

$$(5) \quad \|\mathbf{u}\|_{L^\infty(0,T;L^2(\Omega_F(t))) \cap L^2(0,T;H^1(\Omega_F(t)))} \leq c(\mathbf{f})$$

and

$$(6) \quad \|\mathbf{d}\|_{W^{1,\infty}(0,T;L^2(\Omega_S^0)) \cap L^\infty(0,T;Y)} \leq c(\mathbf{f})$$

Actually it is the kind of stability that we want to preserve in the spatial discretization. To start with, we discretize the reference configuration Ω_S^0 with an appropriate finite element mesh of size h and associate an appropriate finite element space Y_h . We have now to determine a discretization associated to the fluid part. There are many ways to proceed. Here we shall view the domain $\Omega_F(t)$ as the range, through some (time dependent) one to one mapping $\Phi(t)$, of some domain $\hat{\Omega}_F$ (e.g. the initial domain $\Omega_F(0)$ and we shall take this example hereafter). We assume also

that the boundary of the domain $\hat{\Omega}_F$ is composed of the structure reference domain $\Omega_S(0)$ and a part $\hat{\Gamma}_0$ associated to the fixed portion Γ_0 . We shall mesh $\hat{\Omega}_F$ (with a triangulation of size H) and define over this (fluid) reference domain an acceptable couple (\hat{X}_H, \hat{M}_H) of finite element spaces for the approximation of the Stokes problem (we refer to [5] or [2] for more about this question). We then use the mapping $\Phi(t)$ to define the mesh and the appropriate spaces over $\Omega_F(t)$. The major question is then : how is defined the mapping from $\hat{\Omega}_F$ onto $\Omega_F(t)$?

Of course, it has to coincide in some sense with $\mathbf{x} + \mathbf{d}_h(\mathbf{x}, t)$. Hence we define an operator π_H^* that will allow to associate to each $\mathbf{x} + \mathbf{d}_h(\mathbf{x}, t)$ a discrete position of the interface $\pi_H^*(\mathbf{x} + \mathbf{d}_h(\mathbf{x}, t))$ adapted to the mesh (of size H) that exists on the side $\partial\hat{\Omega}_F \setminus \hat{\Gamma}_0$. From this position $\pi_H^*(\mathbf{x} + \mathbf{d}_h(\mathbf{x}, t))$ extended to $\hat{\Gamma}_0$ by the identity (that is thus only given over the boundary of $\hat{\Omega}_F$) we define, by prolongation, a mapping $\Phi_H(t)$ that is a finite element function over the mesh of $\hat{\Omega}_F(0)$. This mapping provides, at the same time, the domain $\Omega_{H,F}(t)$, the mesh on this domain and the spaces of discretization $(X_H(t))$ and $(M_H(t))$ and finally the velocity of the mesh \mathbf{u}_H^* that verifies

$$(7) \quad \mathbf{u}_H^*(\pi_H^*(\mathbf{x} + \mathbf{d}_h(\mathbf{x}, t)), t) = \pi_H^*\left(\frac{\partial \mathbf{d}_h}{\partial t}(\mathbf{x}, t)\right)$$

In order to define the discrete problem associated to (4), we first give the proper interpretation at the discrete level of the equality between the velocity of the fluid and the velocity of the structure. Of course, the equality all over the interface cannot be exactly satisfied, this is why we have to introduce again a projection operator π_H from the h mesh onto the H one. We introduce the discrete equivalent of V as follows

$$V_{h,H}(t) = \{(\mathbf{v}_H, \mathbf{b}_h) \in X_H(t) \times Y_h, \quad \mathbf{v}_H(\pi_h^*(\mathbf{x} + \mathbf{d}(\mathbf{x}, t)), t) = \pi_H(\mathbf{b}_h(\mathbf{x}, t))\}$$

and we look for a solution $(\mathbf{u}_H, p_H, \mathbf{d}_h) \in X_H(t) \times M_H(t) \times Y_h$ such that

$$(8) \quad \begin{aligned} & \int_{\Omega_{H,F}(t)} \frac{\partial \mathbf{u}_H}{\partial t} \mathbf{v}_H + \nu \int_{\Omega_{H,F}(t)} \nabla \mathbf{u}_H \nabla \mathbf{v}_H + \frac{1}{2} \int_{\Omega_{H,F}(t)} \mathbf{u}_H \cdot \nabla \mathbf{u}_H \cdot \mathbf{v}_H - \\ & \frac{1}{2} \int_{\Omega_{H,F}(t)} \mathbf{u}_H \cdot \nabla \mathbf{v}_H \cdot \mathbf{u}_H + \int_{\Gamma_H(t)} \frac{\mathbf{u}_H(t) \mathbf{v}_H(t)}{2} \mathbf{u}_H^* \cdot \mathbf{n} + \int_{\Omega_{H,F}(t)} p_H \nabla \cdot \mathbf{v}_H + \\ & \int_{\Omega_S^0} \frac{\partial^2 \mathbf{d}_h}{\partial t^2} \mathbf{b}_h + a(\mathbf{d}_h, \mathbf{b}_h) = \int_{\Omega_{H,F}(t)} \mathbf{f} \mathbf{v}_H, \quad \forall (\mathbf{v}_H, \mathbf{b}_h) \in V_{h,H}(t), \\ & \int_{\Omega_{H,F}(t)} \nabla \cdot \mathbf{u}_H \mu_H = 0, \quad \forall \mu_H \in M_H(t), \end{aligned}$$

and of course complemented with the coupling condition

$$(9) \quad \mathbf{u}_H(\pi_h^*(\mathbf{x} + \mathbf{d}(\mathbf{x}, t)), t) = \pi_H\left(\frac{\partial \mathbf{d}_h(\mathbf{x}, t)}{\partial t}\right)$$

Note that we have chosen here, as is often the case, to treat the nonlinear convection terms in a skew symmetric way.

It is an easy matter to note that $(\mathbf{u}_H, \frac{\partial \mathbf{d}_h}{\partial t})$ is an admissible test function since from (9), it satisfies the coupling condition on the interface. By plugging this choice of test functions in equation (8) and using the discrete incompressibility condition

in (8) we first get

$$\begin{aligned} \int_{\Omega_{H,F}(t)} \frac{1}{2} \frac{\partial \mathbf{u}_H^2}{\partial t} + \nu \int_{\Omega_{H,F}(t)} (\nabla \mathbf{u}_H)^2 + \int_{\Gamma_H(t)} \frac{\mathbf{u}_H^2(t)}{2} \mathbf{u}_H^* \cdot \mathbf{n} \\ + \int_{\Omega_S^0} \frac{\partial^2 \mathbf{d}_h}{\partial t^2} \frac{\partial \mathbf{d}_h}{\partial t} + a(\mathbf{d}_h, \frac{\partial \mathbf{d}_h}{\partial t}) = \int_{\Omega_{H,F}(t)} \mathbf{f} \mathbf{u}_H, \end{aligned}$$

Reminding the Taylor derivation theorem about integral derivatives, and recalling that the velocity of the interface is \mathbf{u}_H^* , we end at

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \int_{\Omega_{H,F}(t)} \mathbf{u}_H^2 + \nu \int_{\Omega_{H,F}(t)} (\nabla \mathbf{u}_H)^2 \\ + \frac{1}{2} \frac{d}{dt} \int_{\Omega_S^0} \left(\frac{\partial \mathbf{d}_h}{\partial t} \right)^2 + \frac{1}{2} \frac{d}{dt} a(\mathbf{d}_h, \mathbf{d}_h) \leq \|\mathbf{f}\|_{L^2(\Omega_{H,F}(t))} \|\mathbf{u}_H\|_{L^2(\Omega_{H,F}(t))} \end{aligned}$$

which, similarly as in the continuous case (5),(6) leads to a stability result (and thus to an existence result on any fixed time) of the discrete solution

$$(10) \quad \begin{aligned} \|\mathbf{u}_H\|_{L^\infty(0,T;H^1(\Omega_{H,F}(t))) \cap H^1(0,T;L^2(\Omega_{H,F}(t)))} \\ + \|\mathbf{d}_h\|_{L^\infty(0,T;H^1(\Omega_S^0) \cap W^{1,\infty}(0,T;L^2(\Omega_S^0)))} \leq C(\mathbf{f}) \end{aligned}$$

Now that this problem is set, we want to understand how to define the operator π_H so as to obtain an optimal error that could read

$$(11) \quad \begin{aligned} \|\mathbf{u} - \mathbf{u}_H\|_{L^2(0,T;H^1)} + \|p - p_H\|_{L^2(0,T;L^2)} + \|\mathbf{d} - \mathbf{d}_h\|_{L^\infty(0,T;H^1)} \\ \leq c \inf_{\mathbf{v}_H} \|\mathbf{u} - \mathbf{v}_H\|_{L^2(0,T;H^1)} + \inf_{q_H} \|p - q_H\|_{L^2(0,T;L^2)} + \inf_{\mathbf{b}_h} \|\mathbf{d} - \mathbf{b}_h\|_{L^\infty(0,T;H^1)}. \end{aligned}$$

This error analysis is far out of hand since currently, as far as we know, no general existence result is available on the continuous coupled problem in the general case (see however [6]). Nevertheless, since this discretization question is mainly related to the spatial discretization, we present in the following section the numerical analysis of a simplified steady problem in which we believe that all the main features for the definition of the coupling operator π_H are present.

3. Steady State Study

We shall degenerate here the original problem as follows:

- the problem is steady
- the “fluid equations” are replaced by a Laplace equation
- the structure has only a normal displacement that is modelled through a fourth order equation

We consider the problem where a Laplace equation is set on an unknown domain that is delimited over an edge, the equation of which is determined through a fourth order equation in the right hand side of which stands the normal derivative of the solution to the Laplace equation. We denote by $\hat{\Omega} =]0, 1[^2$ the unit square of \mathbb{R}^2 . We take two given functions \mathbf{f} and g respectively in $(L^2(\mathbb{R}^2))^2$ and $H^{-2}(0, 1)$.

We are looking for $\mathbf{v} = (v_1, v_2) \in (H_0^1(\varphi(d)(\hat{\Omega})))^2$ such that :

$$(12) \quad \begin{cases} -\Delta \mathbf{v} = \mathbf{f} & \text{in } \varphi(d)(\hat{\Omega}), \\ \mathbf{v} = 0 & \text{over } \partial \varphi(d)(\hat{\Omega}), \end{cases}$$

and $d \in H_0^2(0, 1)$ such that

$$(13) \quad \begin{cases} \frac{d^4 d}{dx^4} &= g - ((\nabla \mathbf{v}) \circ \varphi(d)) \operatorname{cof} \nabla \varphi(d) \cdot \mathbf{n}_2 \text{ sur } (0, 1), \\ \frac{d}{dx} d(1) &= \frac{d}{dx} d(0) = d(1) = d(0) = 0, \end{cases}$$

where $\varphi(d)$ maps $\hat{\Omega}$ onto $\varphi(d)(\hat{\Omega})$ and is one to one and satisfies on the interface

$$(14) \quad \varphi(d)(x, 1) = (x, 1 + d(x)).$$

A simple choice for $\varphi(d)$ is the following

$$(15) \quad \varphi(d)(x, y) = (x, y + yd(x)).$$

We can rewrite this strong formulation in a variational formulation. We introduce the space of test functions

$$V_d = \left\{ (\mathbf{w}, b) \in H_0^1(\varphi(d)(\hat{\Omega})) \times H_{0,\Gamma_0}^1(\varphi(d)(\hat{\Omega})) \times H_0^2(0, 1) / w_2 \circ \varphi(d) = b \text{ sur }]0, 1[\times \{1\} \right\}.$$

The problem is then the following :

find $(\mathbf{v}, d) \in (H_0^1(\varphi(d)(\hat{\Omega})))^2 \times H_0^2(0, 1)$ such that

$$(16) \quad \begin{cases} \int_{\varphi(d)(\hat{\Omega})} \nabla \mathbf{v} \nabla \mathbf{w} + \int_0^1 \frac{d^2 d}{dx^2} \frac{d^2 b}{dx^2} = \int_{\varphi(d)(\hat{\Omega})} \mathbf{f} \mathbf{w} + \langle g, b \rangle_{H^{-2}, H_0^2} \\ \forall (\mathbf{w}, b) \in V_d. \end{cases}$$

Let ε and α be two real numbers, $0 < \varepsilon < 1$ and $0 < \alpha < 1/2$. We search the displacement in the set defined by

$$B_\varepsilon^\alpha = \left\{ z \in H_0^{2-\alpha}(0, 1) / \|z\|_{H_0^{2-\alpha}(0, 1)} \leq M^{-1}(1 - \varepsilon) \right\},$$

where M is the continuity constant of the injection of $H^{2-\alpha}(0, 1)$ in $C^1([0, 1])$. The problem (16) has at least a solution, for small enough exterior forces. We have

THEOREM 1. Assume that \mathbf{f} and g satisfy

$$\|g\|_{H^{-2}(0, 1)} + C(\varepsilon) \|\mathbf{f}\|_{(L^2(\mathbb{R}^2))^2} \leq M^{-1}(1 - \varepsilon).$$

Then there exists a solution of (16), $(\mathbf{v}, d) \in (H_0^1(\varphi(d)(\hat{\Omega})))^2 \times (H_0^2(0, 1) \cap B_\varepsilon^\alpha)$. If we suppose, moreover, that the function \mathbf{f} is lipschitz with an L^2 -norm and a lipschitz constant small enough, then the solution is unique.

The proof of this theorem is based on a fixed point theorem. For a given deformation γ of the interface Γ , we have the existence of $(\mathbf{v}(\gamma), d(\gamma))$ and we prove that the application T defined by :

$$\begin{aligned} T : B_\varepsilon^\alpha &\rightarrow T(B_\varepsilon^\alpha) \\ \gamma &\mapsto d(\gamma), \end{aligned}$$

satisfies the hypothesis of Schauder theorem.

Next, we want to discretize the problem. For the fluid part, we consider a P_k finite element discretization, with $k \geq 1$, we denote by H the associated space step and X_H the associated finite element space. This discretization of the reference domain $\hat{\Omega}$ is mapped on the deformed configuration as was explained in section 2. For the structure part, we consider P_3 - Hermitian finite element, since the

displacement is solution of a fourth-order equation. The space step is denoted by h and Y_h^0 is the associated finite element space.

To discretize the problem we are going to work with the variational formulation but written on the reference domain $\hat{\Omega}$. With the help of the mapping $\varphi(d)$ we can change of variables in (16). We obtain

$$(17) \quad \begin{aligned} & \int_{\hat{\Omega}} \nabla \varphi(d)^{-t} \nabla \varphi(d)^{-1} \det(\nabla \varphi(d)) \nabla(\mathbf{u}) \nabla \mathbf{w} + \int_0^1 \frac{d^2 d}{dx^2} \frac{d^2 b}{dx^2} = \\ & \int_{\hat{\Omega}} \mathbf{f} \circ \varphi(d) \det(\nabla \varphi(d)) \mathbf{w} + \langle g, b \rangle_{H^{-2}, H_0^2}, \quad \forall (\mathbf{w}, b) \in V^*. \end{aligned}$$

where

$$V^* = \left\{ (\mathbf{w}, b) \in H_0^1(\hat{\Omega}) \times H_{0,\Gamma_0}^1(\hat{\Omega}) \times H_0^2(0, 1) / \mathbf{w}_2 = b \text{ sur } \Gamma \right\},$$

and $\mathbf{v} \circ \varphi(d) = \mathbf{u}$. We set $F(\gamma) = \nabla \varphi(\gamma)^{-t} \nabla \varphi(\gamma)^{-1} \det(\nabla \varphi(\gamma))$.

The discrete variational formulation is the following find $\mathbf{u}_H \in (X_H^0)^2$ and $d_h \in Y_h^0$ such that

$$(18) \quad \begin{cases} \int_{\hat{\Omega}} \mathbf{F}(d_h) \nabla \mathbf{u}_H \nabla \mathbf{w}_H + \int_0^1 \frac{d^2 d_h}{dx^2} \frac{d^2 b_h}{dx^2} = \\ \langle g, b \rangle_{H^{-2}, H_0^2} + \int_{\hat{\Omega}} (1 + d_h) \mathbf{f} \circ \varphi(d_h) \mathbf{w}_H, \quad \forall (\mathbf{w}_H, b_h) \in V_{H,h}, \end{cases}$$

with

$$\begin{aligned} X_H &\stackrel{\text{def}}{=} \{ \mathbf{v} \in C^0(\overline{\hat{\Omega}}) / \mathbf{v}|_T \in P_k(T), \forall T \in \tau_H \}, \\ Y_h^0 &\stackrel{\text{def}}{=} \{ b \in C^1([0, 1]) / b|_S \in P_3(S), \forall S \in \tau_h \} \cap H_0^2(0, 1), \\ X_H^0 &\stackrel{\text{def}}{=} \{ \mathbf{v} \in C^0(\overline{\hat{\Omega}}) / \mathbf{v}|_T \in P_k(T), \forall T \in \tau_H \} \cap H_0^1(\hat{\Omega}), \\ V_{H,h} &\stackrel{\text{def}}{=} \{ (\mathbf{w}_H, b_h) \in (X_H)^2 \times Y_h^0 / \mathbf{w}_H|_{\Gamma_0} = 0, (\mathbf{w}_H)_2|_{\Gamma} = \Pi_H(b_h), \\ &\quad (\mathbf{w}_H)_1 \in X_H^0 \}. \end{aligned}$$

and τ_H (resp. τ_h) denotes the triangulation associated to the fluid part (resp. to the structure) and Π_H represents the matching operator of the test functions. As was explained in the previous section, the nonconforming grids prevent the discrete test functions to satisfy the continuity condition at the interface. The discrete space of test functions $V_{H,h}$ is thus not included in the continuous space V . We are going to study different type of matching : a pointwise matching and an integral matching. On one hand, we will consider for Π_H the finite element interpolation operator associated to the fluid part, and in the other hand the mortar finite element operator [1]. So, we have the two cases

$$(19) \quad \begin{aligned} V_{H,h} &= \{ (\mathbf{w}_H, b_h) \in (X_H)^2 \times Y_h^0 / \mathbf{w}_H|_{\Gamma_0} = 0, (\mathbf{w}_H)_2|_{\Gamma} = I_H(b_h), \\ &\quad (\mathbf{w}_H)_1 \in X_H^0 \}, \end{aligned}$$

where I_H denotes the finite element interpolation operator, and

$$(20) \quad \begin{aligned} V_{H,h} &= \{ (\mathbf{w}_H, b_h) \in (X_H)^2 \times Y_h^0 / \mathbf{w}_H|_{\Gamma_0} = 0, (\mathbf{w}_H)_1|_{\Gamma} = 0, \\ &\quad \int_0^1 (b_h - (\mathbf{w}_H)_2) \psi = 0 \forall \psi \in \tilde{X}_H(\Gamma) \}, \end{aligned}$$

where $\tilde{X}_H(\Gamma)$ is a subspace of codimension 2 in $X_H(\Gamma)$, space of trace on Γ of X_H and define by

$$\tilde{X}_H(\Gamma) \stackrel{\text{def}}{=} \{ w_h \in X_H(\Gamma) / \forall T \in \tau_H, \text{ if } (0, 1) \in T \text{ or if } (1, 1) \in T, w_h|_{\bar{\Gamma} \cap T} \in P_{k-1}(T) \}.$$

THEOREM 2. Let $\mathbf{f} \in (W^{2,\infty}(\mathbb{R}^2))^2$ and $g \in H^{-2}(0, 1)$ there exists $(\mathbf{u}_H, d_h) \in X_H^0 \times Y_h^0$ solution of (18) such that

For the pointwise matching through the interpolation operator I_H , we have

- if $k \leq 2$

$$\begin{aligned}\|d - d_h\|_{H_0^2(0,1)} &\leq C(\mathbf{f}, \lambda) \left[H^k + \|d - d_h^*\|_{H_0^2(0,1)} \right], \\ \|\mathbf{u} - \mathbf{u}_H\|_{(H_0^1(\hat{\Omega}))^2} &\leq C(\mathbf{f}, \lambda) \left[H^k + \|d - d_h^*\|_{H_0^2(0,1)} \right].\end{aligned}$$

- if $k > 2$

$$\begin{aligned}\|d - d_h\|_{H_0^2(0,1)} &\leq C(\mathbf{f}, \lambda) \left[H^2 + \|d - d_h^*\|_{H_0^2(0,1)} \right], \\ \|\mathbf{u} - \mathbf{u}_H\|_{(H_0^1(\hat{\Omega}))^2} &\leq C(\mathbf{f}, \lambda) \left[H^2 + \|d - d_h^*\|_{H_0^2(0,1)} \right].\end{aligned}$$

For the matching through the mortar operator, we have

$$\begin{aligned}\|d - d_h\|_{H_0^2(0,1)} &\leq C(\mathbf{f}, \lambda) \left[H^k + \|d - d_h^*\|_{H_0^2(0,1)} \right], \\ \|\mathbf{u} - \mathbf{u}_H\|_{(H_0^1(\hat{\Omega}))^2} &\leq C(\mathbf{f}, \lambda) \left[H^k + \|d - d_h^*\|_{H_0^2(0,1)} \right].\end{aligned}$$

where d_h^* denotes the projection of d on Y_h^0 in semi norm $H^2(0, 1)$.

We remark that for the finite element interpolation operator we obtain optimal error estimates when the degree of the fluid polynomial is less or equal to 2. These estimates are no more optimal when $k \geq 2$. This is due to the fact that the displacement is solution of a fourth order equation. When the weak matching is imposed through the mortar operator then the error estimates are optimal in all the cases.

The proof of this result is based on a discrete fixed point theorem due to Brezzi Rappaz Raviart in a modified version due to Crouzeix [4]. This theorem gives us the existence of the discrete solution together with the error between this discrete solution and the exact solution. We want to underline the reason why the pointwise matching yields optimal error estimates in some (interesting) cases. When we consider the nonconforming discretization of a second order equation using nonoverlapping domain decomposition for the pointwise matching the error estimates are never optimals. In fact in both situations, the error analysis involves a best fit error and a consistency error which measures the effect of the nonconforming discretization. Classically, for the Laplace equation (c.f. Strang's lemma) this term can be written as follows

$$\sup_{w_h \in V_\delta} \frac{\int_{\Gamma} \frac{\partial u}{\partial n} [w_\delta]}{\|w_\delta\|_{H^{1/2}(\Gamma)}},$$

where $[w_\delta]$ represents the jump at the interface of the functions belonging to the discrete space V_δ , and u denotes the exact solution. In the fluid structure interaction, even if we deal with a non linear problem a similar term appears (not exactly under this form) in the proof and affects the final estimate. The jump of the test functions at the interface is equal to $\mathbf{w}_H - b_h = \Pi_H(b_h) - b_h$. Since b_h is $H^2(0, 1)$, for $\Pi_H = I_H$ we have

$$\|I_H(b_h) - b_h\|_{L^2(0,1)} \leq CH^2 \|b_h\|_{H^2(0,1)}.$$

TABLE 1

	“1D Structure ” second order operator	“1D Structure ” fourth order operator	“2D Structure ” (Laplace equation)
P_k 2D “Fluid” + interpolation	$k = 1$ optimal $k > 1$ non optimal	$k \leq 2$ optimal $k > 2$ non optimal	non optimal
P_k 2D “Fluid” + Mortar Method	optimal $\forall k$	optimal $\forall k$	optimal $\forall k$

That explains why for $k \leq 2$ the estimates is optimal. On the opposite the integral matching (mortar element method) gives for all value of k optimal error estimates. We have also studied a linear problem in the case where the displacement is solution of a second order equation on the interface (this is the case when the longitudinal displacements are taken into account). We can summarise the results in Table 1. For more details see [7].

Acknowledgment The authors have appreciated the discussions with Charbel Farhat.

References

1. C. Bernardi, Y. Maday, and A. T. Patera, *A new non conforming approach to domain decomposition : the mortar element method*, Collège de France Seminar, Pitman, H Brezis, J.L. Lions (1990).
2. F. Brezzi and Michel Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New-York NY, 1991.
3. D. Errate, M. Esteban, and Y. Maday, *Couplage fluide structure, un modèle simplifié*, Note aux C.R.A.S. **318** (1994).
4. M. Crouzeix et J. Rappaz, *On numerical approximation in bifurcation theory*, Masson, Paris, Milan, Barcelone, 1989.
5. V. Girault et P.A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, 1986.
6. C. Grandmont et Y. Maday, *Analysis of a 2d fluid structure interaction*, in preparation (1998).
7. C. Grandmont, *Analyse mathématique et numérique de quelques problèmes d'interactions fluide structure*, PhD dissertation, University Pierre et Marie Curie, 4, place Jussieu, Paris, January 1998.
8. C. Grandmont, V. Guimet, and Y. Maday, *Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid-structure interaction*, in preparation (1998).
9. Lessoine and C. Farhat, *Stability analysis of dynamic meshes for transient aeroelastic computations*, 11th AIAA Computational Fluid Dynamics Conference, Orlando, Florida (1993).
10. J. Mouro and P. Le Tallec, *Structure en grands déplacement couplées à des fluides en mouvement*, Rapport INRIA N°2961 (1996).
11. S. Piperno, C. Farhat, and B. Larrouyrou, *Partitioned procedures for the transient solution of coupled aeroelastic problems. Part I : Model problem, theory, and two-dimensional application*, Comp. Methods in Appl. Mech. and Eng. **124** (1995).

LABORATOIRE D'ANALYSE NUMÉRIQUE, UNIVERSITÉ PIERRE ET MARIE CURIE, 4, PLACE JUSSIEU, 75252 PARIS CEDEX 05, FRANCE

E-mail address: grandmon@ann.jussieu.fr

LABORATOIRE D'ANALYSE NUMÉRIQUE, UNIVERSITÉ PIERRE ET MARIE CURIE, 4, PLACE JUSSIEU, 75252 PARIS CEDEX 05, FRANCE AND LABORATOIRE ASCI, BAT. 506, UNIVERSITÉ PARIS SUD 91405 ORSAY, CEDEX, FRANCE

E-mail address: maday@ann.jussieu.fr

Hash-Storage Techniques for Adaptive Multilevel Solvers and Their Domain Decomposition Parallelization

Michael Griebel and Gerhard Zumbusch

1. Introduction

Partial differential equations can be solved efficiently by adaptive multigrid methods on a parallel computer. We report on the concepts of hash-table storage techniques and space-filling curves to set up such a code. The hash-table storage requires substantial less amount of memory and is easier to code than tree data structures used in traditional adaptive multigrid codes, already for the sequential case. The parallelization takes place by a domain decomposition by space filling curves, which are intimately connected to the hash table. The new data structure simplifies the parallel version of the code substantially and introduces a cheap way to solve the load balancing and mapping problem.

We study a simple model problem, an elliptic scalar differential equation on a two-dimensional domain. A finite difference discretization of the problem leads to a linear equation system, which is solved efficiently by a multigrid method. The underlying grid is adapted in an iterative refinement procedure. Furthermore, we run the code on a parallel computer. In the overall approach we then put all three methods (multigrid, adaptivity, parallelism) efficiently together.

While state-of-the-art computer codes use tree data structures to implement such a method, we propose hash tables instead. Hash table addressing gives more or less direct access to the data stored (except of the collision cases), i.e. it is proven to possess a $\mathcal{O}(1)$ complexity with a moderate constant if a statistical data distribution is assumed. Hash tables allow to deal with locally adapted data in a simple way. Furthermore, data decomposition techniques based on space-filling curves provide a simple and efficient way to partition the data and to balance the computational load.

We demonstrate the concepts of hash-storage and space-filling curves by a simple example code, using a square shaped two-dimensional domain and finite difference discretization of the Laplacian. The concepts can also be applied to more complicated domains, equations and grids. A finite element discretization on an unstructured tetrahedral grid for example requires more data, more complicated data structures and more lines of code. However, the concepts presented in this article remain attractive even for such a code.

1991 *Mathematics Subject Classification*. Primary 65Y05; Secondary 65N55, 65N06, 65N50.

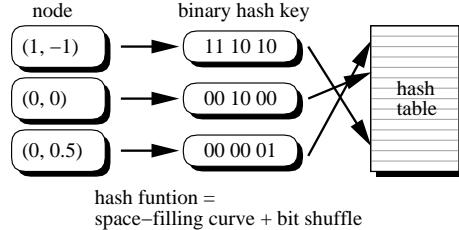


FIGURE 1. Storing nodes in a hash table.

2. Hash Addressing

2.1. Hash-Storage. Looking for a different way to manage adaptive grids than tree data structures, we propose to use hash storage techniques. *Hash tables* are a well established method to store and retrieve large amounts of data, see f.e. [8, chap. 6.4]. They are heavily used in database systems, computer language interpreters such as ‘Perl’ and the Unix ‘C shell’ and in compilers. We propose to use hash table for numerics.

The idea of hashing is to map each entity of data to a *hash-key* by a hash-function. The hash-key is used as an address in the hash table. The entity is stored and can be retrieved at that address in the hash table, which is implemented as a linear vector of cells (buckets) as illustrated in Figure 1. Since there are many more possible different entities than different hash-keys, the hash function cannot be injective. Algorithms to resolve collisions are needed. Furthermore, some buckets in the hash table may be left empty, because no present entity is mapped to that key. We use space-filling curves as hash functions, see Chapter 3.

In general, access to a specific entry in the hash table can be performed in constant time, which is cheaper than random access in a sorted list or a tree. However, this is only true if the hash function scatters the entries broad enough and there are enough different cells in the hash table.

The hash table code does not need additional storage overhead for logical connectivities like tree-type data structures which are usually used in adaptive finite element codes [9]. Furthermore, and this is an additional advantage of the hash table methodology, it allows relatively easy coding and parallelization with simple load balancing.

2.2. Finite Difference Discretization. We take a strictly node-based approach. The nodes are stored in a hash table. Each interior node represents one unknown. Neither elements nor edges are stored. We use a one-irregular grid with ‘hanging’ nodes, see Figure 2, whose values are determined by interpolation. This is equivalent to the property that there is at most one ‘hanging’ node per edge. The one-irregular condition is a kind of a geometric smoothness condition for the adaptive grid. Additionally we consider only square shaped elements.

The partial differential equation is discretized by finite differences. We set up the operator as a set of difference stencils from one node to its neighboring nodes in the grid, which can be easily determined: Given a node, its neighbors can be only on a limited number of level, or one level up or down. The distance to the neighbor is determined by the level they share.

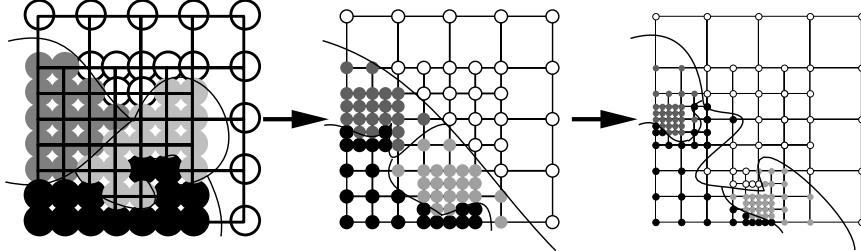


FIGURE 2. A Sequence of adaptively refined grids mapped onto four processors.

So pure geometric information is sufficient to apply the finite difference operator to some vector. We avoid the storage of the stiffness matrix or any related information. For the iterative solution of the equation system, we have to implement matrix multiplication, which is to apply the operator to a given vector. A loop over all nodes in the hash table is required for this purpose.

2.3. Multilevel Preconditioner. We use an additive version of the multigrid method for the solution of the equation system, i.e. the so called BPX preconditioner [5].

$$Bu = \sum_{\text{level } j} \sum_i 4^{-j}(u, \phi_i^j) \phi_i^j$$

This requires an outer Krylov iterative solver. The BPX preconditioner has the advantage of an optimal $\mathcal{O}(1)$ condition number and an implementation of order $\mathcal{O}(n)$, which is optimal, even in the presence of degenerate grids. Furthermore, this additive version of multigrid is also easier to parallelize than multiplicative multigrid versions.

The straightforward implementation is similar to the implementation of a multi-grid V-cycle. However, the implementation with optimal order is similar to the hierarchical basis transformation and requires one auxiliary vector. Two loops over all nodes are necessary, one for the restriction operation and one for the prolongation operation. They can be both implemented as a tree traversal. However, by iterating over the nodes in the right order, two ordinary loops over all nodes in the hash table are sufficient, one forward and one backward.

2.4. Adaptive Refinement. In order to create adaptive grids, we have to locate areas, where to refine the grid. Applying an error estimator or error indicator gives an error function defined on the grid. With some threshold value, the estimated error is converted into a flag field, determining whether grid refinement is required in the neighborhood. Then, large error values result in refinement. In the next step, new nodes are created. Finally a geometric grid has to be constructed, which fulfills the additionally imposed geometric constraints, e.g. one-irregularity.

3. Space-Filling Curves

3.1. Space-Filling Curve Enumeration. A domain or data decomposition is needed for the parallelization of the code. Usually domain decomposition and mapping strategies are difficult and expensive. Adaptive grids require such an

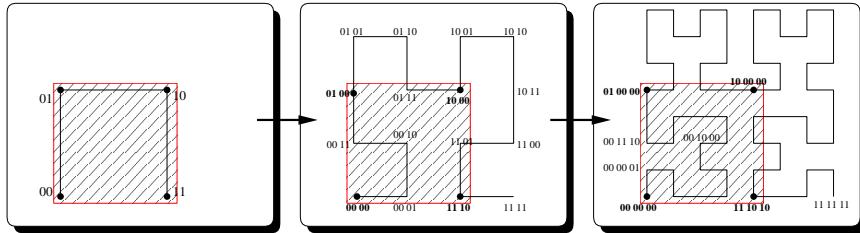


FIGURE 3. Hilbert's space-filling curve at different levels of resolution. It covers the whole domain, say $\Omega = [-1, 1]^2$ (drawn shaded). The nodes are numbered binary from 0 to $4^j - 1$. Additionally, the numbers are used for the hash-keys.

expensive decomposition several times. Hence we are interested in cheap decomposition heuristics.

We choose a computational very cheap method based on space-filling curves [14]. A space-filling curve is defined recursively by substituting a straight line segment by a certain pattern of lines, similar to fractals. This recursion is applied infinitely times and results in a curve, which fills the whole domain. The curve defines a (continuous) mapping from an interval to the whole domain via a scaled arc-length. Space-filling curves are often used for theoretical purposes, e.g. for complexity bounds. Furthermore, they have been employed in combinatorial optimization [1], in computer graphics [16], in operating systems and routing, and in parallel computing as a tool for domain partitioning [4, 12].

We use space-filling curves as a way to enumerate and order nodes in the computational domain. One can think of such a space-filling curve as passing all nodes of a given grid, e.g. an adaptive grid. Because of the boundary nodes, we choose a curve which covers a larger domain than the computational domain, see Figure 3. We assign the scaled arc length of the curve to each node of the grid, called index. The indices imply a total order relation on the nodes. The space-filling curve is never constructed explicitly, but it is used for the computation of the indices. The indices are used for the construction of hash-keys.

3.2. Space-Filling Curve Partition. Given an ordered list of nodes induced by a space-filling curve, we construct a static partition of the grid points and data in the following way: We cut the list into p equally sized intervals and map them according to this order to processors with increasing numbers. The partition is defined by its $p - 1$ cuts.

The computational load is balanced exactly, see [18, 13]. The volume of communication depends on the boundaries of the partitions.

4. Parallel Code

For the parallelization of the sequential code, all its components such as the solution of the linear system, the estimation of errors and the creation of nodes have to be done in parallel. Additionally the data has to be distributed to the processors. This is done in a load balancing and mapping step right after creating new nodes, a step which was not present in the sequential version or for uniform refinement [7]. We consider a distributed memory, MIMD, message passing paradigm. This makes

the parallelization more involved than it would be on a shared memory computer as in [9, 3]. Parallelizing a tree based code is quite complicated and time consuming. Here, algorithms must be implemented on sub-trees. Furthermore, algorithms for moving and for joining sub-trees must be implemented. Finally all this must be done in a consistent and transparent way, as indicated in [19, 17, 2, 15, 10]. However, the parallelization of an adaptive code based on hash tables, which we consider here, will turn out to be much easier.

4.1. Partition in Parallel. Using the space filling curve, the partitioning problem reduces to a sorting problem. This requires a parallel sort algorithm with distributed input and output. We employ a one-stage *radix sort* algorithm, see [8, chap. 5.2.5]. Here we can make use of the assumption that the previous data decomposition still guarantees good load-balancing for the parallel sort.

The result is a new partition of the grid. In total, the space-filling curve load balancing is very cheap, because most of the data has been sorted in a previous step. It parallelizes very well and thus can be applied in each step of the computation.

The index of a node induced by the space-filling curve is used for assigning the node to a processor and additionally for addressing the node in the local hash table of the processor. In case that a copy of a node (a ghost node) is required on another processor, the index is also used for addressing the copy in the hash table of this processor. Comparing the index of a node to the $p - 1$ partition cut values, it is easy to determine the processor the node originally belongs to.

4.2. Finite Differences in Parallel. The parallel iterative solution consists of several components. The Krylov iterative solver requires matrix multiplications, scalar products and the application of the BPX-preconditioner in our case. The scalar product can be implemented as ordinary data reduction operations [11] offered by any message passing library. Any modification of a node's value is performed by the processor who owns the node, "owner computes". This implies a rule of how the computational work is partitioned to the processors.

The matrix multiplication requires the update of auxiliary (ghost) values located at the boundary of the partition, see [6]. The variables of ghost nodes in this region are filled with actual values. Then, the local matrix multiplication can take place without any further communication, and only one local nearest neighbor communication is necessary.

4.3. Multilevel Preconditioner in Parallel. The communication pattern of the BPX-preconditioner is more dense than pattern for the matrix multiplication: The local restriction operations can be performed in parallel without any communication. The resulting values have to be reduced [11] and distributed.

The local restriction and prolongation operations are organized as ordinary restriction and prolongation, just restricted to the local nodes and ghost nodes on a processor. They can be implemented either as tree traversals or as a forward and a backward loops on properly ordered nodes, i.e. on the hash table. The ghost nodes are determined as set of ghost nodes of grids on all levels. Hence the communication takes place between nearest neighbors, where neighbors at all grid levels have to be considered. In this sense the communication pattern is between all-to-all and a pure local pattern.

Each node sums up the values of all its distributed copies. This can be implemented by two consecutive communication steps, fetching and distributing the

TABLE 1. Uniform refinement example, timing, levels 6 to 9, 1 to 8 processors.

time		processors			
		1	2	4	8
levels	6	0.0580	0.0326	0.0198	0.0473
	7	0.2345	0.1238	0.0665	0.1861
	8	1.0000	0.4914	0.2519	0.2350
	9			1.1297	0.6282

values. Now the restricted values are present on all nodes and ghost nodes. Finally, the reverse process of prolongation can take place as local operations again. Thus the result is valid on all nodes without the ghost nodes.

Compared to multiplicative multigrid methods where communication on each level takes place separately in the smoothing process, the hierarchical nearest neighbor communication is a great advantage [19]. However, the total volume of data to be communicated in the additive and the multiplicative multigrid method are of the same order (depending on the number of smoothing steps). This means that the additive multigrid has an advantage for computers with high communication latency, while for computers with low latency the number of communication steps is less important.

5. Experiments

We consider the two dimensional Poisson equation $-\Delta u = 0$ on $\Omega = [-1, 1]^2$ with Dirichlet boundary conditions $u = 0$ on $\partial\Omega \setminus [-1, 0]^2$ and $u = 1$ on the remainder of $\partial\Omega$. We run our adaptive multilevel finite difference code to solve it. The solution possesses two singularities located at the jumps in the boundary data $(-1, 0)$ and $(0, -1)$. All numbers reported are scaled CPU times measured on a cluster of SGI O2 workstations, running MPICH on a fast ethernet network.

5.1. Uniform Example. In the first test we consider regular grids (uniform refinement). Table 1 shows wall clock times for the solution of the equation system on a regular grid of different levels using different numbers of processors.

We observe a scaling of a factor of 4 from one level to the next finer level which corresponds to the factor of 4 increase in the amount of unknowns on that level. The computing times decay and a scale-up can be seen. However, the 8 processor perform efficiently only for sufficiently large problems, i.e. for problems with more than 8 levels.

5.2. Adaptive Example. In the next test we consider adaptive refined grids. The grids are refined towards the two singularities. Table 2 depicts times in the adaptive case. These numbers give the wall clock times for the solution of the equation system again, now on different levels of adaptive grids and on different numbers of processors.

We obtain a scaling of about a factor 4 from one level to the next finer level. This is due to an increase of the amount of nodes by a factor of 4, because the grid has been adapted already towards the singularities on previous levels. Increasing the number of processors speeds up the computation accordingly, at least for two and four processors. In order to use seven processor efficiently, the grid has to be fine enough, i.e it has to have more than 8 levels.

TABLE 2. Adaptive refinement example, timing, levels 7 to 9, 1 to 7 processors.

levels	time	processors			
		1	2	4	7
7	0.0578	0.0321	0.0187	0.0229	
8	0.2291	0.1197	0.0645	0.0572	
9	1.0000	0.5039	0.2554	0.1711	

TABLE 3. Ratio sorting nodes (partitioning and mapping) to solving the equation system (multilevel), level 8, 1 to 8 processors.

solve time	sort time	processors			
		1	2	4	7/8
uniform	0	0.0028	0.0079	0.0141	
adaptive	0	0.0066	0.0149	0.2367	

5.3. Load Balancing. Now we compare the time for solving the equation system with the time required for sorting the nodes and mapping them to processors. The ratio indicates how expensive the load balancing and mapping task is in comparison to the rest of the code. We give the values in Table 3 for the previous uniform and adaptive refinement examples using different numbers of processors.

In the single processor case, no load balancing is needed, so the sort time to solve time ratio is zero. In the uniform grid case the numbers stay below two percent. In the adaptive grid case, load balancing generally is more expensive. But note that load balancing still is much cheaper than solving the equation systems. However, higher number of processors make the mapping relatively slower.

In the case of uniform refinement, for a refined grid, there are only few nodes located at processor boundaries which may have to be moved during the mapping. Hence our load balancing is very cheap in this case. Mapping data for adaptive refinement requires the movement of a large amount of data because of the overall amount of data, even if most of the nodes stay on the processor. Other load balancing strategies can be quite expensive for adaptive refinement procedures, see [2].

6. Conclusion

We have introduced hash storage techniques for the solution of partial differential equations by a parallel adaptive multigrid method. Hash tables lead to a substantial reduction of memory requirements to store sequences of adaptive grids compared to standard tree based implementations. Furthermore, the implementation of an adaptive code based on hash tables proved to be simpler than the tree counterpart. Both properties, low amount of memory and especially the simple programming, carried over to the parallelization of the code. Here space filling curves were used for data partitioning and at the same time for providing a proper hash function.

The results of our numerical experiments showed that load balancing based on space filling curves is indeed cheap. Hence we can in fact afford to use it in each grid refinement step. Thus our algorithm operates on load balanced data at

any time. This is in contrary to other procedures, which have to be used often in connection with more expensive load balancing mechanisms, where load imbalance is accumulated for several steps.

References

1. J. Bartholdi and L. K. Platzman, *Heuristics based on space-filling curves for combinatorial optimization problems.*, Management Science **34** (1988), 291–305.
2. P. Bastian, *Parallele adaptive Mehrgitterverfahren*, B. G. Teubner, Stuttgart, 1996.
3. K. Birken, *Ein Parallelisierungskonzept für adaptive, numerische Berechnungen*, Master's thesis, Universität Erlangen-Nürnberg, 1993.
4. S. H. Bokhari, T. W. Crockett, and D. N. Nicol, *Parametric binary dissection*, Tech. Report 93-39, ICASE, 1993.
5. J. H. Bramble, J. E. Pasciak, and J. Xu, *Parallel multilevel preconditioners*, Math. Comp. **55** (1990), 1–22.
6. G. C. Fox, *Matrix operations on the homogeneous machine.*, Tech. Report C3P-005, Caltech, 1982.
7. M. Griebel, *Parallel domain-oriented multilevel methods*, SIAM Journal on Scientific Computing **16** (1995), no. 5, 1105–1125.
8. D. E. Knuth, *The art of computer programming*, vol. 3, Addison-Wesley, 1973.
9. P. Leinen, *Ein schneller adaptiver Löser für elliptische Randwertprobleme auf Seriell- und Parallelrechnern*, Ph.D. thesis, Universität Dortmund, 1990.
10. W. Mitchell, *A parallel multigrid method using the full domain partition*, Electronic Transactions on Numerical Analysis (97), Special issue for proceedings of the 8th Copper Mountain Conference on Multigrid Methods.
11. MPI Forum, *MPI: A message-passing interface standard*, University of Tennessee, Knoxville, Tennessee, 1.1 ed., 1995.
12. J. T. Oden, A. Patra, and Y. Feng, *Domain decomposition for adaptive hp finite element methods*, Proceedings of Domain Decomposition 7, Contemporary Mathematics, vol. 180, AMS, 1994, pp. 295–301.
13. M. Parashar and J.C. Browne, *On partitioning dynamic adaptive grid hierarchies*, Proceedings of the 29th Annual Hawaii International Conference on System Sciences, 1996.
14. H. Sagan, *Space-filling curves*, Springer, 1994.
15. L. Stals, *Parallel multigrid on unstructured grids using adaptive finite element methods*, Ph.D. thesis, Department of Mathematics, Australian National University, 1995.
16. D. Voorhies, *Space-filling curves and a measure of coherence*, Graphics Gems II (J. Arvo, ed.), Academic Press, 1994, pp. 26–30.
17. C. H. Walshaw and M. Berzins, *Dynamic load-balancing for PDE solvers on adaptive unstructured meshes*, Tech. Report Computer Studies 92.32, University of Leeds, 1992.
18. M. Warren and J. Salmon, *A portable parallel particle program*, Comput. Phys. Comm. **87** (1995), 266–290.
19. G. W. Zumbusch, *Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme*, SFB-Report 342/19/91A, TUM-I9127, TU München, 1991.

INSTITUT FÜR ANGEWANDTE MATHEMATIK, UNIVERSITY BONN, WEGELERSTR. 6, 53115
BONN, GERMANY
E-mail address: griebel@iam.uni-bonn.de

INSTITUT FÜR ANGEWANDTE MATHEMATIK, UNIVERSITY BONN, WEGELERSTR. 6, 53115
BONN, GERMANY
E-mail address: zumbusch@iam.uni-bonn.de

Extension of a Coarse Grid Preconditioner to Non-symmetric Problems

Caroline Japhet, Frédéric Nataf, and François-Xavier Roux

1. Introduction

The Optimized Order 2 (OO2) method is a non-overlapping domain decomposition method with differential interface conditions of order 2 along the interfaces which approximate the exact artificial boundary conditions [13, 9]. The convergence of Schwarz type methods with these interface conditions is proved in [12]. There already exists applications of the OO2 method to convection-diffusion equation [9] and Helmholtz problem [3]. We first recall the OO2 method and present numerical results for the convection-diffusion equation discretized by a finite volume scheme. The aim of this paper is then to provide an extension of a preconditioning technique introduced in [7, 5] based upon a global coarse problem to non-symmetric problems like convection-diffusion problems. The goal is to get the independence of the convergence upon the number of subdomains. Numerical results on convection-diffusion equation will illustrate the efficiency of the OO2 algorithm with this coarse grid preconditioner.

2. The Optimized Order 2 Method

We recall the OO2 Method in the case of the convection-diffusion problem:

$$(1) \quad \begin{aligned} L(u) &= cu + a(x, y) \frac{\partial u}{\partial x} + b(x, y) \frac{\partial u}{\partial y} - \nu \Delta u = f \text{ in } \Omega \\ C(u) &= g \text{ on } \partial\Omega \end{aligned}$$

where Ω is a bounded open set of \mathbb{R}^2 , $\vec{a} = (a, b)$ is the velocity field, ν is the viscosity, C is a linear operator, c is a constant which could be $c = \frac{1}{\Delta t}$ with Δt a time step of a backward-Euler scheme for solving the time dependent convection-diffusion problem. The method could be applied to other PDE's.

The OO2 method is based on an extension of the additive Schwarz algorithm with non-overlapping subdomains : $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$, $\Omega_i \cap \Omega_j = \emptyset$, $i \neq j$. We denote by $\Gamma_{i,j}$ the common interface to Ω_i and Ω_j , $i \neq j$. The outward normal from Ω_i is \mathbf{n}_i and $\boldsymbol{\tau}_i$ is a tangential unit vector.

1991 *Mathematics Subject Classification*. Primary 65F30; Secondary 65M60, 65Y05.

The additive Schwarz algorithm with non-overlapping subdomains ([11]) is :

$$(2) \quad \begin{aligned} L(u_i^{n+1}) &= f, \quad \text{in } \Omega_i \\ B_i(u_i^{n+1}) &= B_i(u_j^n), \quad \text{on } \Gamma_{i,j}, \quad i \neq j \\ C(u_i^{n+1}) &= g \quad \text{on } \partial\Omega_i \cap \partial\Omega \end{aligned}$$

where B_i is an interface operator. We recall first the OO2 interface operator B_i and then the substructuring formulation of the method.

2.1. OO2 interface conditions. In the case of Schwarz type methods, it has been proved in [14] that the optimal interface conditions are the exact artificial boundary conditions [8]. Unfortunately, these conditions are pseudo-differential operators. Then, it has been proposed in [13] to use low wave number differential approximations to these optimal interface conditions. Numerical tests on a finite difference scheme with overlapping subdomains has shown that the convergence was very fast for a velocity field non tangential to the interface, but very slow, even impossible, for a velocity field tangential to the interface. So, instead of taking low-wave number approximations, it has been proposed in [9] to use differential interface conditions of order 2 along the interface which optimize the convergence rate of the Schwarz algorithm. These “Optimized Order 2” interface operators are defined as follows:

$$B_i = \frac{\partial}{\partial n_i} - \frac{\mathbf{a} \cdot \mathbf{n}_i - \sqrt{(\mathbf{a} \cdot \mathbf{n}_i)^2 + 4c\nu}}{2\nu} + c_2 \frac{\partial}{\partial \tau_i} - c_3 \frac{\partial^2}{\partial \tau_i^2}$$

where $c_2 = c_2(\mathbf{a} \cdot \mathbf{n}_i, \mathbf{a} \cdot \boldsymbol{\tau}_i)$ and $c_3 = c_3(\mathbf{a} \cdot \mathbf{n}_i, \mathbf{a} \cdot \boldsymbol{\tau}_i)$ minimize the convergence rate of the Schwarz algorithm. The analytic analysis in the case of 2 subdomains and constant coefficients in (1) reduce the minimization problem to a one parameter minimization problem. This technique is extended in the case of variable coefficients and an arbitrary decomposition, that is only one parameter is computed, with a dichotomy algorithm. With this parameter we get c_2 and c_3 (see [10]). So the OO2 conditions are easy to use and not costly. The convergence of the Schwarz algorithm with the OO2 interface conditions is proved for a decomposition in N subdomains (strips) using the techniques in [12].

2.2. Substructuring formulation. In [14], the non-overlapping algorithm (2) is interpreted as a Jacobi algorithm applied to the interface problem

$$(3) \quad D\lambda = b$$

where λ , restricted to Ω_i , represents the discretization of the term $B_i(u_i)$ on the interface $\Gamma_{i,j}$, $i \neq j$. The product $D\lambda$, restricted to Ω_i , represents the discretization of the jump $B_i(u_i) - B_i(u_j)$ on the interface $\Gamma_{i,j}$, $i \neq j$. To accelerate convergence, the Jacobi algorithm is replaced by a Krylov type algorithm [16].

2.3. Numerical results. The method is applied to a finite volume scheme [1] (collaboration with MATRA BAe Dynamics France) with a decomposition in N non-overlapping subdomain. The interface problem (3) is solved by a BICG-stab algorithm. This involves solving N independent subproblems which can be done in parallel. Each subproblem is solved by a direct method. We denote by h the mesh size. We compare the results obtained with the OO2 interface conditions and the Taylor order 0 ([4],[2], [13]) or order 2 interface conditions ([13]).

TABLE 1. Number of iterations versus the convection velocity's angle: 16×1 subdomains, $\nu = 1.d - 2$, $CFL = 1.d9$, $h = \frac{1}{241}$, $\log_{10}(Error) < 1.d - 6$

convection velocity	OO2	Taylor order 2	Taylor order 0
normal velocity to the interface $a = y, b = 0$	15	123	141
tangential velocity to the interface $a = 0, b = y$	20	not convergent	75

TABLE 2. Number of iterations versus the mesh size: 16×1 subdomains, $a = y, b = 0, \nu = 0.01, CFL = 1.d9, \log_{10}(Error) < 1.d - 6$

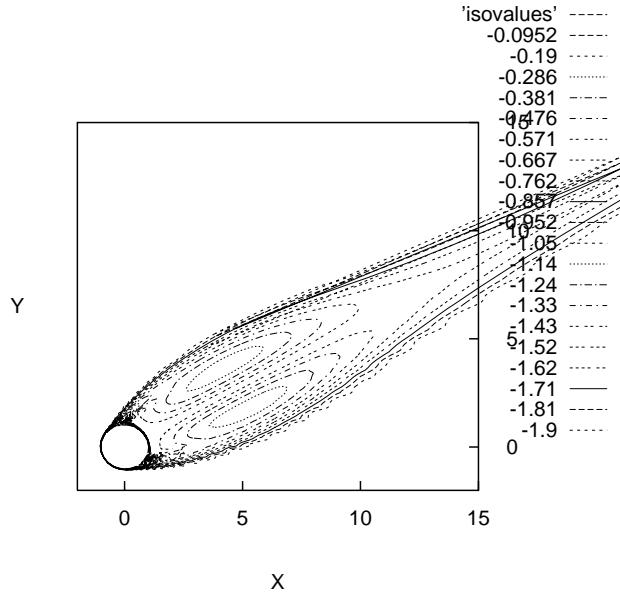
grid	65×65	129×129	241×241
OO2	15	15	15
Taylor order 2	49	69	123
Taylor order 0	49	82	141

TABLE 3. Number of iterations versus the mesh size: 16×1 subdomains, rotating velocity, $a = -\sin(\pi(y - \frac{1}{2}))\cos(\pi(x - \frac{1}{2})), b = \cos(\pi(y - \frac{1}{2}))\sin(\pi(x - \frac{1}{2}))$, $\nu = 1.d - 2, CFL = 1.d9, \log_{10}(Error) < 1.d - 6$

grid	65×65	129×129	241×241
OO2	49	48	48
Taylor order 0	152	265	568

1. We consider the problem: $L(u) = 0, 0 \leq x \leq 1, 0 \leq y \leq 1$ with $u(0, y) = \frac{\partial u}{\partial x}(1, y) = 0, 0 \leq y \leq 1, \frac{\partial u}{\partial y}(x, 1) = 0, u(x, 0) = 1, 0 \leq x \leq 1$. In order to observe the influence on the convergence both of the convection velocity angle to the interfaces, and of the mesh size, we first take a decomposition in strips. The Table 1 shows that the OO2 interface conditions give a significantly better convergence which is independent of the convection velocity angle to the interfaces. One of the advantages is that for a given number of subdomains, the decomposition of the domain doesn't affect the convergence. We also observe that the convergence for the studied numerical cases is independent of the mesh size (see Table 2 and Table 3).

2. The OO2 method was also tested for a convection velocity field issued from the velocity field of a Navier-Stokes incompressible flow, with Reynolds number $Re = 10000$, around a cylinder. This velocity field is issued from a computation at the aerodynamic department at Matra. The computational domain is defined by $\Omega = \{(x, y) = (r \cos(\theta), r \sin(\theta)), 1 \leq r \leq R, 0 \leq \theta \leq 2\pi\}$ with $R > 0$ given. We consider the problem $L(u) = 0$ in Ω with $u = 1$ on $\{(x, y) = (\cos(\theta), \sin(\theta)), 0 \leq \theta \leq 2\pi\}$ and $u = 0$ on $\{(x, y) = (R \cos(\theta), R \sin(\theta)) | 0 \leq \theta \leq 2\pi\}$. The grid is $\{(x, y) = (r_i \cos(\theta_j), r_i \sin(\theta_j))\}$, and is refined around the cylinder and in the direction of the flow. We note $N_{max} = (\text{number of points on the boundary of a subdomain}) \times (\text{number of subdomains})$. The OO2 interface conditions give

FIGURE 1. Iso-values of the solution u , $\nu = 1.d - 4$, $CFL = 1.d9$ TABLE 4. Number of iterations versus the viscosity; 4×2 subdomains, $CFL = 1.d9$, $\log_{10}(\text{Error}) < 1.d - 6$

	OO2	Taylor order 2	Taylor order 0
$\nu = 1.d - 5$	56	41	119
$\nu = 1.d - 4$	43	121	374
$\nu = 1.d - 3$	32	$N_{\max} = 768$ $\log_{10}(\text{Error}) = -5.52$	$N_{\max} = 768$ $\log_{10}(\text{Error}) = -2.44$

also significantly better convergence in that case. Numerically the convergence is practically independent of the viscosity ν (see Table 4).

3. Extension of a coarse grid preconditioner to non-symmetric problems

Numerically, the convergence ratio of the method is nearly linear upon the number of subdomains in one direction of space. To tackle this problem, the aim of this paper is to extend a coarse grid preconditioner introduced in [7], [5] to non-symmetric problems like convection-diffusion problems. This preconditioning technique has been introduced for the FETI method, in linear elasticity, when local Neumann problems are used and are ill posed (see [7]). It has been extended for plate or shell problems, to tackle the singularities at interface cross-points ([6], [5],

[15]). In that case, this preconditioner is a projection for $(D.,.)_2$ on the space orthogonal to a coarse grid space which contain the corner modes. This consists in constraining the Lagrange multiplier to generate local displacement fields which are continuous at interface cross-points. The independence upon the number of subdomains has been proved.

In this paper we extend this preconditioner by considering a $(D.,D.)_2$ projection on the space orthogonal to a coarse grid space. The goal is to filter the low frequency phenomena, in order to get the independence of the convergence upon the number of subdomains. So the coarse grid space, denoted W , is a set of functions called “coarse modes” which are defined on the interfaces by :

- Preconditioner M1 : the “coarse modes” are the fields with unit value on one interface and 0 on the others.
- Preconditioner M2 : the “coarse modes” in a subdomain Ω_i are on one interface the restriction of $K_i u_i$ where $u_i = 1 \in \Omega_i$ and K_i is the stiffness matrix, and 0 on the others.

Then, at each iteration, λ^p satisfies the continuity requirement of associated field u^p at interface :

$$(DW)_i^t(D\lambda^p - b) = 0 \quad \forall i$$

That is, if we introduce the projector P on W^\perp for $(D.,D.)_2$, the projected gradient of the condensed interface problem is:

$$(4) \quad Pg^p = g^p + \sum_i (DW)_i \delta_i$$

and verify

$$(5) \quad (DW)_i^t Pg^p = 0 \quad \forall i$$

With (4), the condition (5) can be written as the coarse problem :

$$(DW)^t (DW) \delta = -(DW)^t g^p$$

So the method has two level : at each iteration of the Krylov method at the fine level, an additional problem has to be solved at the coarse grid level.

3.1. Numerical results. The preconditioned OO2 method is applied to problem (1) discretized by the finite volume scheme with non-overlapping subdomains. The interface problem (3) is solved by a projected GCR algorithm, that is the iterations of GCR are in the $(D.,D.)_2$ orthogonal to the coarse grid space. Each subproblem is solved by a direct method. We compare the results obtained with the preconditioners M1 and M2.

1. We consider the problem: $L(u) = 0$, $0 \leq x \leq 1$, $0 \leq y \leq 1$ with $\frac{\partial u}{\partial x}(1, y) = 0$, $u(0, y) = 1$, $0 \leq y \leq 1$ and $\frac{\partial u}{\partial y}(x, 1) = 0$, $u(x, 0) = 1$, $0 \leq x \leq 1$. The convection velocity is $a = y$, $b = 0$. In that case, the solution is constant in all the domain : $u = 1$ in $[0, 1]^2$. Table 5 justify the choice of the preconditioner M2. In fact, in that case the field λ associated to the solution on the interfaces is in the coarse grid space of preconditioner M2.

2. We consider the problem: $L(u) = 0$, $0 \leq x \leq 1$, $0 \leq y \leq 1$ with $\frac{\partial u}{\partial x}(1, y) = u(0, y) = 0$, $0 \leq y \leq 1$ and $\frac{\partial u}{\partial y}(x, 1) = 0$, $u(x, 0) = 1$, $0 \leq x \leq 1$,

TABLE 5. Number of iterations, 8×1 subdomains $a = y$, $b = 0$, $\nu = 1.d - 2$, $CFL = 1.d9$, $h = \frac{1}{129}$, $\log_{10}(Error) < 1.d - 6$

	without preconditioner	preconditioner M1	preconditioner M2
OO2	15	17	1

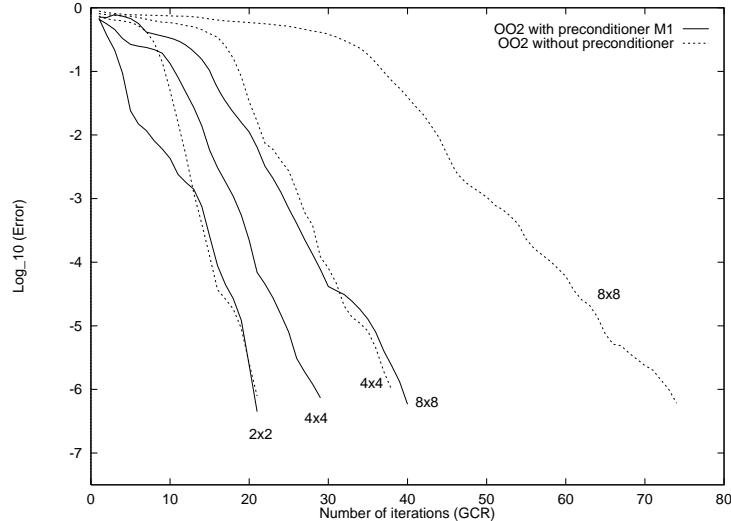


FIGURE 2. Preconditioner M1: Decomposition in $N \times N$ subdomains ($N = 2, 4, 8$); rotating velocity, $\nu = 1.d - 2$, $CFL = 1.d9$, $h = \frac{1}{241}$

with a rotating convection velocity: $a = -\sin(\pi(y - \frac{1}{2}))\cos(\pi(x - \frac{1}{2}))$ and $b = \cos(\pi(y - \frac{1}{2}))\sin(\pi(x - \frac{1}{2}))$. Different methods have been developed to solve this problem (see for example [17]). Here we want to observe the behavior of the preconditioner on this problem. Figure 3 shows that the convergence of the OO2 method with the preconditioner M2 is nearly independent of the number of subdomains. The convergence is better with preconditioner M2 than preconditioner M1 (figure 2).

4. Conclusion

The OO2 method appears to be a very efficient method, applied to convection-diffusion problems. With the coarse grid preconditioner, the convergence ratio is numerically nearly independent of the number of subdomains.

References

1. C. Borel and M. Bredif, *High performance parallelized implicit Euler solver for the analysis of unsteady aerodynamic flows*, First European CFD Conference, Brussels (1992).
2. C. Carlenzoli and A. Quarteroni, *Adaptive domain decomposition methods for advection-diffusion problems*, Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations (I. Babuska and Al., eds.), The IMA Volumes in Mathematics and its applications, no. 75, Springer-Verlag, 1995, pp. 165–187.

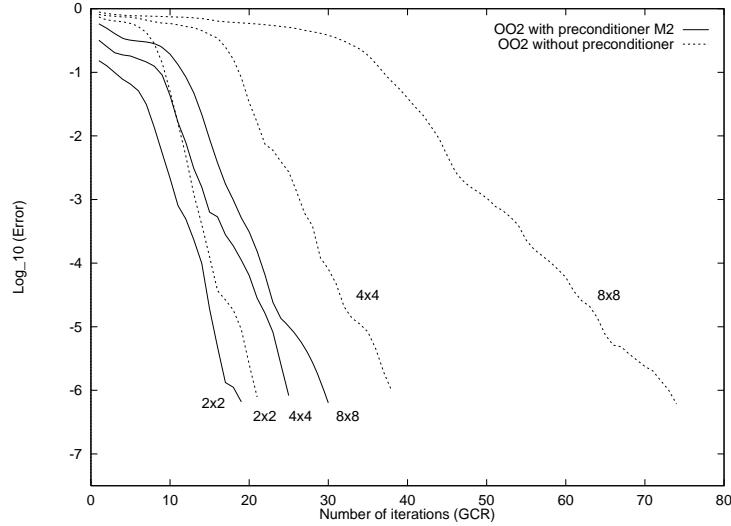


FIGURE 3. Preconditioner M2: Decomposition in $N \times N$ subdomains ($N = 2, 4, 8$); rotating velocity, $\nu = 1.d - 2$, $CFL = 1.d9$, $h = \frac{1}{241}$

3. P. Chevalier and F. Nataf, *Une méthode de décomposition de domaine avec des conditions d'interface optimisées d'ordre 2 (OO2) pour l'équation d'Helmholtz*, note CRAS (1997), (To appear).
4. B. Despres, *Décomposition de domaine et problème de Helmholtz*, C.R. Acad. Sci., Paris **311** (1990), 313–316.
5. C. Farhat, P.S. Chen, J. Mandel, and F.X. Roux, *The two-level FETI method - Part II: Extension to shell problems, parallel implementation and performance results*, Computer Methods in Applied Mechanics and Engineering, (in press).
6. 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, (in press).
7. C. Farhat, J. Mandel, and F.X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Computer Methods in Applied Mechanics and Engineering **115** (1994), 367–388.
8. L. Halpern, *Artificial boundary conditions for the advection-diffusion equations*, Math. Comp. **174** (1986), 425–438.
9. C. Japhet, *Optimized Krylov-Ventcell method. Application to convection-diffusion problems*, Nine International Conference on Domain Decomposition Methods for Partial Differential Equations (M. Espedal P. Bjorstad and D. Keyes, eds.), John Wiley & Sons Ltd, (in press).
10. ———, *Conditions aux limites artificielles et décomposition de domaine : Méthode OO2 (Optimisé Ordre 2). Application à la résolution de problèmes en mécanique des fluides*, Rapport interne 373, CMAP, Ecole Polytechnique, October 1997.
11. P. L. Lions, *On the Schwarz alternating method III: A variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, 1989, pp. 202–223.
12. F. Nataf and F. Nier, *Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains*, Numerische Mathematik **75** (1997), 357–377.
13. F. Nataf and F. Rogier, *Factorisation of the convection-diffusion operator and the Schwarz algorithm*, *M³AS* **5** (1995), no. 1, 67–93.
14. F. Nataf, F. Rogier, and E. de Sturler, *Domain decomposition methods for fluid dynamics, Navier-Stokes equations on related non linear analysis* (A. Sequeira, ed.), Plenum Press Corporation, 1995, pp. 307–377.

15. F.X. Roux and C Farhat, *Parallel implementation of the two-level FETI method*, Nine International Conference on Domain Decomposition Methods for Partial Differential Equations (M. Espedal P. Bjorstad and D. Keyes, eds.), John Wiley & Sons Ltd, (in press).
16. Y. Saad, *Iterative methods for sparse linear systems*, PWS Publishing Compagny, 1996.
17. W.P. Tang, *Numerical solution of a turning point problem*, Fifth International Conference on Domain Decomposition Methods for Partial Differential Equations (T. Chan, D. Keyes, G. Meurant, S. Scroggs, and R. Voigt, eds.), Norfolk, 1991.

ONERA, DTIM/CHP, 29 AVENUE DE LA DIVISION LECLERC, BP 72, 92322 CHÂTILLON CEDEX, FRANCE

E-mail address: japhet@onera.fr

CMAP, CNRS URA756, ECOLE POLYTECHNIQUE, 91128 PALAISEAU CEDEX, FRANCE

E-mail address: nataf@cmapx.polytechnique.fr

ONERA, DTIM/CHP, 29 AVENUE DE LA DIVISION LECLERC, BP 72, 92322 CHÂTILLON CEDEX, FRANCE

E-mail address: roux@onera.fr

On the Interaction of Architecture and Algorithm in the Domain-based Parallelization of an Unstructured-grid Incompressible Flow Code

Dinesh K. Kaushik, David E. Keyes, and Barry F. Smith

1. Introduction

The convergence rates and, therefore, the overall parallel efficiencies of additive Schwarz methods are often notoriously dependent on subdomain granularity. Except when effective coarse-grid operators and intergrid transfer operators are known, so that optimal multilevel preconditioners can be constructed, the number of iterations to convergence and the communication overhead per iteration tend to increase with granularity for elliptically-controlled problems, for either fixed or memory-scaled problem sizes.

In practical large-scale applications, however, the convergence rate degradation of fine-grained single-level additive Schwarz is sometimes not as serious as the scalar, linear elliptic theory would suggest. Its effects are mitigated by several factors, including pseudo-transient nonlinear continuation and dominant intercomponent coupling that can be captured exactly in a point-block ILU preconditioner. We illustrate these claims with encouraging scalabilities for a legacy unstructured-grid Euler flow application code, parallelized with the pseudo-transient Newton-Krylov-Schwarz algorithm using the PETSc library. We note some impacts on performance of the horizontal (distributed) and vertical (hierarchical) aspects of the memory system and consider architecturally motivated algorithmic variations for their amelioration.

2. Newton-Krylov-Schwarz

The discrete framework for an implicit PDE solution algorithm, with pseudotimestepping to advance towards an assumed steady state, has the form: $(\frac{1}{\Delta t^\ell})\mathbf{u}^\ell + \mathbf{f}(\mathbf{u}^\ell) = (\frac{1}{\Delta t^\ell})\mathbf{u}^{\ell-1}$, where $\Delta t^\ell \rightarrow \infty$ as $\ell \rightarrow \infty$. Each member of the sequence of nonlinear problems, $\ell = 1, 2, \dots$, is solved with an inexact Newton method. The

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65Y05, 65N30.

Supported in part by NASA contract NAGI-1692 and by a GAANN fellowship from the U.S. Department of Education.

Supported in part by the National Science Foundation grant ECS-9527169 and by NASA Contracts NAS1-19480 and NAS1-97046.

Supported by U.S. Department of Energy, under Contract W-31-109-Eng-38.

resulting Jacobian systems for the Newton corrections are solved with a Krylov method, relying only on matrix-vector multiplications. 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 parallelize well already, 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. The true inverse of the Jacobian is usually dense, reflecting the global Green’s function of the continuous linearized PDE operator it approximates. A good preconditioner saves time and space by permitting fewer iterations in the Krylov loop and smaller storage for the Krylov subspace. An additive Schwarz preconditioner [4] accomplishes this in a localized manner, with an approximate solve in each subdomain of a partitioning of the global PDE domain. Applying any preconditioner in an additive Schwarz manner tends to increases flop 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) [5, 8].

When nested within a pseudo-transient continuation scheme to globalize the Newton method [11], the implicit framework (called Ψ NKS) has four levels:

```

do l = 1, n_time
    SELECT TIME-STEP
    do k = 1, n_Newton
        compute nonlinear residual and Jacobian
        do j = 1, n_Krylov
            do i = 1, n_Precon
                solve subdomain problems concurrently
            enddo
            perform Jacobian-vector product
            ENFORCE KRYLOV BASIS CONDITIONS
            update optimal coefficients
            CHECK LINEAR CONVERGENCE
        enddo
        perform vector update
        CHECK NONLINEAR CONVERGENCE
    enddo
enddo

```

The operations written in uppercase customarily involve global synchronizations.

We have experimented with a number of Schwarz preconditioners, with varying overlap and varying degrees of subdomain fill-in, including the new, communication-efficient, Restricted Additive Schwarz (RAS) method [6]. For the cases studied herein, we find the degenerate block Jacobi form with block ILU(0) on the subdomains is adequate for near scalable convergence rates.

3. Parallel Implementation Using PETSc

The parallelization paradigm we employ in approaching a legacy code is a compromise between the “compiler does all” and the “hand-coded by expert” approaches. We employ the “Portable, Extensible Toolkit for Scientific Computing” (PETSc) [2, 3], 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 sub-tasks 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.

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 line-search 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 features relevant to computational fluid dynamicists, including matrix-free Krylov methods, blocked forms of parallel preconditioners, and various types of time-stepping.

A diagram of the calling tree of a typical Ψ NKS application appears below. The arrows represent calls that cross the boundary between application-specific code and PETSc library code; all internal details of both are suppressed. The top-level user routine performs I/O related to initialization, restart, and post-processing and calls PETSc subroutines to create data structures for vectors and matrices and to initiate the nonlinear solver. PETSc calls user routines for function evaluations $\mathbf{f}(\mathbf{u})$ and (approximate) Jacobian evaluations $\mathbf{f}'(\mathbf{u})$ at given vectors \mathbf{u} representing the discrete state of the flow. Auxiliary information required for the evaluation of \mathbf{f} and $\mathbf{f}'(\mathbf{u})$ that is not carried as part of \mathbf{u} is communicated through PETSc via a user-defined “context” that encapsulates application-specific data. (Such information typically includes dimensioning data, grid data, physical parameters, and quantities that could be derived from the state \mathbf{u} , but are most conveniently stored instead of recalculated, such as constitutive quantities.)

4. Parallel Port of an NKS-based CFD Code

We consider parallel performance results for a NASA unstructured grid CFD code that is used to study the high-lift, low-speed behavior of aircraft in take-off and landing configurations. FUN3D [1] is a tetrahedral vertex-centered unstructured grid code developed by W. K. Anderson of the NASA Langley Research Center for compressible and incompressible Euler and Navier-Stokes equations. FUN3D uses a control volume discretization with variable-order Roe schemes for approximating the convective fluxes and a Galerkin discretization for the viscous terms. Our parallel experience with FUN3D is with the incompressible Euler subset thus far,

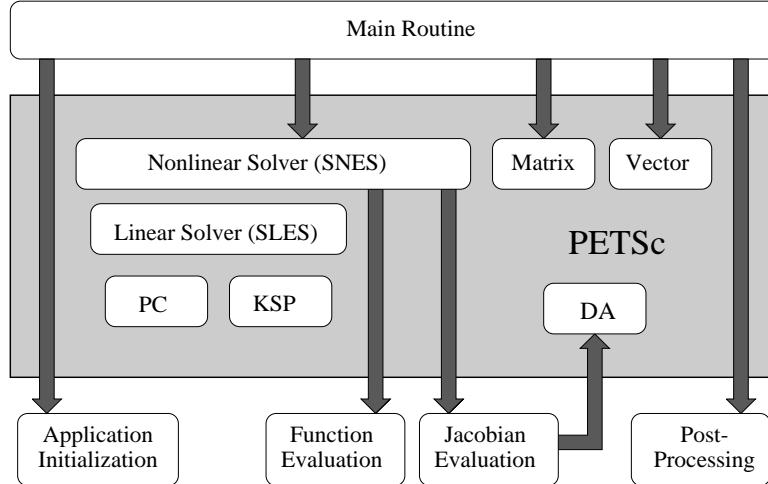


FIGURE 1. Coarsened calling tree of the FUN3D-PETSc code, showing the user-supplied main program and callback routines for providing the initial nonlinear iterate, computing the nonlinear residual vector at a PETSc-requested state, and evaluating the Jacobian (preconditioner) matrix.

but nothing in the solution algorithms or software changes for the other cases. Of course, convergence rate will vary with conditioning, as determined by Mach and Reynolds numbers and the correspondingly induced grid adaptivity. Furthermore, robustness becomes more of an issue in problems admitting shocks or making use of turbulence models. The lack of nonlinear robustness is a fact of life that is largely outside of the domain of parallel scalability. In fact, when nonlinear robustness is restored in the usual manner, through pseudo-transient continuation, the conditioning of the linear inner iterations is enhanced, and parallel scalability may be improved. In some sense, the Euler code, with its smaller number of flops per point per iteration and its aggressive trajectory towards the steady state limit may be a *more*, not less, severe test of scalability.

We employ Ψ NKS with point-block ILU(0) on the subdomains. The original code possesses a pseudo-transient Newton-Krylov solver already. Our reformulation of the global point-block ILU(0) of the original FUN3D into the Schwarz framework of the PETSc version is the primary source of additional concurrency. The timestep grows from an initial CFL of 10 towards infinity according to the switched evolution/relaxation (SER) heuristic of Van Leer & Mulder [12]. In the present tests, the maximum CFL is 10^5 . The solver operates in a matrix-free, split-discretization mode, whereby the Jacobian-vector operations required by the GMRES [13] Krylov method are approximated by finite-differenced Fréchet derivatives of the nonlinear residual vector. The action of the Jacobian is therefore always “fresh.” However, the submatrices used to construct the point-block ILU(0) factors on the subdomains as part of the Schwarz preconditioning are based on a lower-order discretization than the one used in the residual vector, itself. This is a common approach in practical codes, and the requisite distinctions within the residual and Jacobian subroutine calling sequences are available in the legacy FUN3D version.

TABLE 1. Cray T3E parallel performance (357,900 vertices)

procs	its	time	speedup	Efficiency		
				η_{alg}	η_{impl}	$\eta_{overall}$
16	77	2587.95s	1.00	1.00	1.00	1.00
32	75	1262.01s	2.05	1.03	1.00	1.03
64	75	662.06s	3.91	1.03	0.95	0.98
128	82	382.30s	6.77	0.94	0.90	0.85

4.1. Parallel Scaling Results. We excerpt from a fuller report to appear elsewhere [10] tables for a 1.4-million degree-of-freedom (DOF) problem, converged with a relative steady-state residual reduction of 10^{-10} in approximately 6.5 minutes using approximately 1600 global fine-grid flux balance operations (or “work units” in the multigrid sense), on 128 processors of a T3E; and for an 11.0-million DOF problem, converged in approximately 30 minutes on 512 processors. Relative efficiencies in excess of 80% are obtained over relevant ranges of processor number in both cases. Similar results are presented in [10] for the IBM SP. The minimum relevant number of processors is (for our purposes) the smallest power of 2 that can house a problem completely in distributed DRAM. In practice, using fewer than this holds high performance resources captive to paging off of slow disks (and dramatically inflates subsequent parallel speedups!). The maximum relevant number is the maximum number available or the largest power of 2 that allows enough volumetric work per processor to cover the surfacial overhead. In practice, tying up more processors than this for long runs can be construed as wasting DRAM.

The physical configuration is a three-dimensional ONERA M6 wing up against a symmetry plane (see Fig. 2) an extensively studied standard case. Our tetrahedral Euler grids were generated by D. Mavriplis of ICASE. We use a maximum Krylov dimension of 20 vectors per pseudo-timestep. The pseudo-timestepping is a nontrivial feature of the algorithm, since the norm of the steady state residual does not decrease monotonically in the larger grid cases. (In production, we would employ mesh sequencing so that the largest grid case is initialized from the converged solution on a coarser grid. In the limit, such sequencing permits the finer grid simulation to be initialized within the domain of convergence of Newton’s method.)

Table 1 shows a relative efficiency of 85% over the relevant range for a problem of $4 \times 357,900$ DOFs. Each iteration represents one pseudo-timestep, including one Newton correction, and up to 20 Schwarz-preconditioned GMRES steps. Convergence is defined as a relative reduction in the norm of the steady-state nonlinear residual of the conservation laws by a factor of 10^{-10} . The convergence rate typically degrades slightly as number of processors is increased, due to introduction of increased concurrency in the preconditioner, which is partition-dependent, in general.

The overall efficiency, $\eta_{overall}$, is the speedup divided by the processor ratio. The algorithmic efficiency, η_{alg} , is the ratio of iterations to convergence, as processor number varies. The implementation efficiency, η_{impl} , the quotient of $\eta_{overall}$ and η_{alg} , therefore represents the efficiency on a *per iteration* basis, isolated from the slight but still significant algorithmic degradation. η_{impl} is useful in the quantitative understanding of parallel overhead that arises from communication, redundant computation, and synchronization.

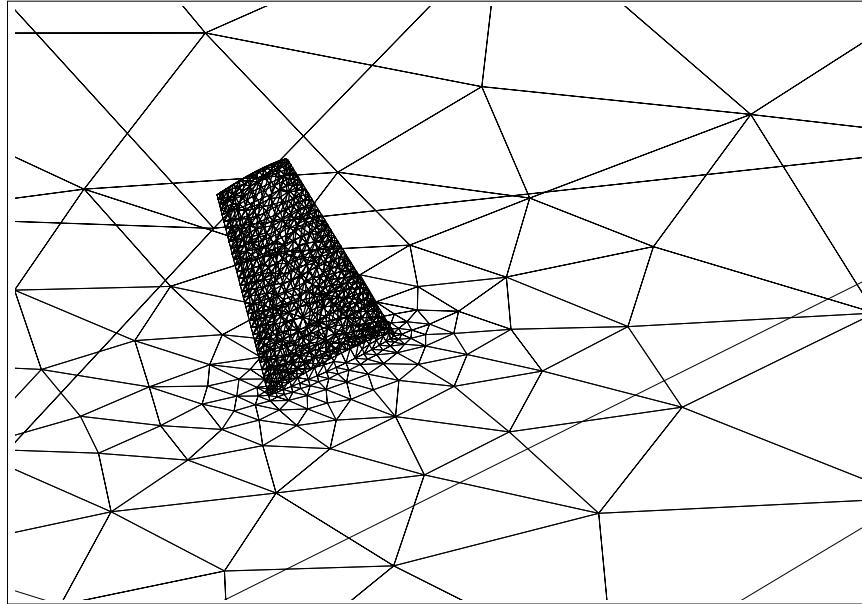


FIGURE 2. *The surface triangulation of the M6 wing, the symmetry root plane, and the farfield bounding surface are shown for a relatively coarse grid.*

TABLE 2. Cray T3E parallel performance (2,761,774 vertices)

procs	its	time	speedup	Efficiency			Gflop/s
				η_{alg}	η_{impl}	$\eta_{overall}$	
128	164	6,048.37s	1.00	1.00	1.00	1.00	8.5
256	166	3,242.10s	1.87	0.99	0.94	0.93	16.6
512	171	1,811.13s	3.34	0.96	0.87	0.83	32.1

Table 2 shows a relative efficiency of 83% over the relevant range for a problem of $4 \times 2,761,774$ DOFs. Each iteration represents up to 45 preconditioned GMRES iterations (with restarting every 16 iterations). This grid is the largest yet generated by our colleagues at NASA Langley for an implicit wing computation. Coordinate and index data (including 18 million edges) alone occupy an 857 MByte file.

The 32.1 Gflop/s achieved on 512 nodes for this sparse unstructured computation is 12% of the best possible rate of 265 Gflop/s for the dense LINPACK benchmark on a matrix of order 79,744 (with 6.36 billion nonzeros) on the identical configuration [7]. The principal “slow” routines (at present) are orthogonalization in the Krylov method and subdomain Jacobian preconditioner formation (soon to be addressed), together accounting for about 20–23% of execution time and running at only 20% of the overall sustained Gflop/s rate.

It is interesting to note the source of the degradation of η_{impl} in going from 128 to 512 processors, since much finer granularities will be required in ASCI-scale computations. The maximum over all processors of the time spent at global

TABLE 3. Cray T3E parallel performance — Gustafson scaling

vert	procs	vert/proc	its	time	time/it
357,900	80	4474	78	559.93s	7.18s
53,961	12	4497	36	265.72s	7.38s
9,428	2	4714	19	131.07s	6.89s

synchronization points (reductions — mostly inner products and norms) is 12% of the maximum over all processors of the wall-clock execution time. This is almost entirely idle time arising from load imbalance, not actual communication time, as demonstrated by inserting barriers before the global reductions and noting that the resulting fraction of wall-clock time for global reductions drops below 1%. Closer examination of partitioning and profiling data shows that although the distribution of “owned” vertices is nearly perfectly balanced, and with it the “useful” work, the distribution of ghosted nodes can be very imbalanced, and with it, the overhead work and the local communication requirements. In other words, the partitioning objective of minimizing total edges cut while equidistributing vertices does *not*, in general, equidistribute the execution time between synchronization points, mainly due to the skew among the processors in ghost vertex responsibilities. This example of the necessity of supporting multiple objectives (or multiple constraints) in mesh partitioning has been communicated to the authors of major partitioning packages, who have been hearing it from other sources, as well. For PDE codes amenable to per-iteration communication and computation work estimates that are not data-dependent, it is plausible to approximately balance multiple distinct phases in an *a priori* partitioning. More generally, partitionings may need to be rebalanced dynamically, on the basis of real-time measurements rather than models. This will require integration of load balancing routines with the solution routines in parallel. We expect that a similar computation after such higher level needs are accommodated in the partitioner will achieve close to 95% overall efficiency on 512 nodes.

Since we possess a sequence of unstructured Euler grids for the same wing, we can perform a Gustafson-style scalability study by varying the number of processors and the discrete problem dimension in proportion. We note that the concept of Gustafson-style scalability does not extend perfectly cleanly to nonlinear PDEs, since added resolution brings out added physics and (generally) poorer conditioning, which may cause a shift in the “market basket” of kernel operations as the work in the nonlinear and linear phases varies. However, our shockless Euler simulation is a reasonably clean setting for this study, if corrected for iteration count. Table 3 shows three computations on the T3E over a range of 40 in problem and processor size, while maintaining approximately 4,500 vertices per processor.

The good news in this experiment is contained in the final column, which shows the average time per parallelized pseudo-transient NKS outer iteration for problems with similarly sized local workingsets. Less than a 7% variation in performance occurs over a factor of nearly 40 in scale.

4.2. Serial Cache Optimization Results. From a processor perspective we have so far looked outward rather than inward. Since the aggregate computational rate is a product of the concurrency and the rate at which computation occurs in

TABLE 4. IBM P2SC cache performance in serial (22,677 vertices)

	Enhancements			Results			
	Field Interlacing	Structural Blocking	Edge Reordering	its	time	time/it	impr. ratio
1				28	2905s	103.8s	—
2	×			25	1147s	45.9s	2.26
3	×	×		25	801s	32.0s	3.24
4	×		×	25	673s	26.9s	3.86
5	×	×	×	25	373s	14.9s	6.97

a single active thread, we briefly discuss the per-node performance of the legacy and the PETSc ported codes. Table 4 shows the effect, individually or in various combinations, of three cache-related performance enhancements, relative to the original vector-oriented code, whose performance is given in row 1. Since the number of iterations differs slightly in the independent implementations, we normalize the execution time by the number of iterations for the comparisons in the final column. These optimizations are described in more detail in [9]. We observe certain synergisms in cache locality; for instance, adding structural blocking to the interlaced code without edge-reordering provides a factor of 1.43, while adding structural blocking to the edge-reordered code provides a factor of 1.81. Similarly, adding edge-reordering to a code without structural blocking provides a factor of 1.71, while adding structural edge-reordering to the blocked code provides a factor of 2.15. Including the iteration count benefit, the cache-oriented serial code executes 7.79 times faster than the original, before parallelization.

5. Conclusions

We have demonstrated very respectable scaling behavior for a Ψ NKS version of a 3D unstructured CFD code. We began with a legacy vector-oriented code known to be algorithmically competitive with multigrid in 2D, improved its performance as far as we could for a sequential cache orientation, and then parallelized it with minimal impact on the sequential convergence rate. The parallel version can be scaled to accommodate very rich grids.

Profiling the highest granularity runs reveals certain tasks that need additional performance tuning — load balancing being the least expected. With respect to the interaction of algorithms with applications we believe that the ripest remaining advances are interdisciplinary: ordering, partitioning, and coarsening must adapt to coefficients (and thus grid spacing, flow magnitude, and flow direction) for convergence rate improvement. Trade-offs between grid sequencing, pseudo-time iteration, nonlinear iteration, linear iteration, and preconditioner iteration must be further understood and exploited.

Acknowledgements

The authors thank W. Kyle Anderson of the NASA Langley Research Center for providing FUN3D. Satish Balay, Bill Gropp, and Lois McInnes of Argonne National Laboratory co-developed (with Smith) the PETSc software employed in this paper. Computer time was supplied by NASA (under the Computational Aero Sciences

section of the High Performance Computing and Communication Program), and DOE (through Argonne and NERSC).

References

1. W. K. Anderson, *FUN2D/3D*, <http://fmad-www.larc.nasa.gov/~wanderso/Fun/fun.html>, 1997.
2. S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object-oriented numerical software libraries*, Modern Software Tools in Scientific Computing, Birkhauser, 1997, pp. 163–201.
3. ———, *The Portable, Extensible Toolkit for Scientific Computing, version 2.0.21*, <http://www.mcs.anl.gov/petsc>, 1997.
4. X.-C. Cai, *Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations*, Tech. report, Courant Institute, NYU, 1989.
5. X.-C. Cai, D. E. Keyes, and V. Venkatakrishnan, *Newton-Krylov-Schwarz: An implicit solver for CFD*, Proceedings of the Eighth International Conference on Domain Decomposition Methods, Wiley, 1997, pp. 387–400.
6. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for nonsymmetric linear systems*, Tech. Report CU-CS-843-97, Computer Science Dept., Univ. of Colorado at Boulder, August 1997, http://www.cs.colorado.edu/cai/public_html/papers/ras_v0.ps.
7. J. J. Dongarra, *Performance of various computers using standard linear equations software*, Tech. report, Computer Science Dept., Univ. of Tennessee, Knoxville, 1997.
8. W. D. Gropp, L. C. McInnes, M. D. Tidriri, and D. E. Keyes, *Parallel implicit PDE computations: Algorithms and software*, Proceedings of Parallel CFD'97, Elsevier, 1998.
9. D. K. Kaushik, D. E. Keyes, and B. F. Smith, *Cache optimization in multicomponent unstructured-grid implicit CFD codes*, (in preparation), 1998.
10. ———, *Newton-Krylov-Schwarz parallelization of unstructured-grid legacy CFD codes using PETSc*, (in preparation), 1998.
11. C. T. Kelley and D. E. Keyes, *Convergence analysis of pseudo-transient continuation*, SIAM J. Num. Anal. (to appear), 1998.
12. W. Mulder and B. Van Leer, *Experiments with implicit upwind methods for the Euler equations*, J. Comp. Phys. **59** (1995), 232–246.
13. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **7** (1986), 856–869.

COMPUTER SCIENCE DEPARTMENT, OLD DOMINION UNIVERSITY, NORFOLK, VA 23529-0162
AND ICASE, NASA Langley Res. Ctr., HAMPTON, VA 23681-2199

E-mail address: kaushik@cs.odu.edu

COMPUTER SCIENCE DEPARTMENT, OLD DOMINION UNIVERSITY, NORFOLK, VA 23529-0162
AND ICASE, NASA Langley Res. Ctr., HAMPTON, VA 23681-2199

E-mail address: keyes@icase.edu

MATHEMATICS AND COMPUTER SCIENCE DIVISION, ARGONNE NATIONAL LABORATORY AR-
GONNE, IL 60439-4844

E-mail address: bsmith@mcs.anl.gov

Additive Domain Decomposition Algorithms for a Class of Mixed Finite Element Methods

Axel Klawonn

1. Introduction

We discuss three different domain decomposition methods for saddle point problems with a penalty term. All of them are based on the overlapping additive Schwarz method. In particular, we present results for a mixed formulation from linear elasticity which is well suited for almost incompressible materials. The saddle point problems are discretized by mixed finite elements; this results in the solution of large, indefinite linear systems. These linear systems are solved by appropriate Krylov space methods in combination with domain decomposition preconditioners. First, we discuss an indefinite preconditioner which can be formulated as an overlapping Schwarz method analogous to the methods for symmetric positive definite problems proposed and analyzed by Dryja and Widlund [3]. The second approach can be interpreted as an inexact, overlapping additive Schwarz method, i.e. a domain decomposition method with inexact subdomain solves. This preconditioner is symmetric positive definite and can be analyzed as a block-diagonal preconditioner, cf. [5]. Our third method is based on a block-triangular formulation, cf. [4, 7], it uses an overlapping additive Schwarz method for each of the block solvers. Numerical results indicate that all of our methods are scalable. For brevity, for a list of references to other domain decomposition approaches for saddle point problems, we refer to Klawonn and Pavarino [6]. There are several other approaches to solve saddle point problems iteratively, for a list of references we refer to [5], [4], and [8].

The results in this paper have been obtained in joint work with Luca F. Pavarino from the University of Pavia, Italy.

The outline of this paper is as follows. In Section 2, we introduce the saddle point problem and a suitable finite element discretization. In Section 3, we present our preconditioner for saddle point problems with a penalty term. In Section 4, computational results are given.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 73C35.

This work has been supported in part by the DAAD (German Academic Exchange Service) within the program HSP III (Gemeinsames Hochschulsonderprogramm III von Bund und Ländern).

2. A mixed formulation of linear elasticity

Let $\Omega \subset \mathbf{R}^d$, $d = 2, 3$ be a polygonal (resp. polyhedral) domain. We consider a linear elastic material and denote by λ and μ the Lamé constants. The linear strain tensor ε is defined by $\varepsilon_{ij} := \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$. The material is assumed to be fixed along the part of the boundary $\Gamma_0 \subset \partial\Omega$, to be subject to a surface force f_1 along $\Gamma_1 := \partial\Omega \setminus \Gamma_0$, and to an external force f_0 . Other parameters often used in the literature are Young's modulus E and the Poisson ratio ν . They are related to the Lamé constants by $\nu = \frac{\lambda}{2(\lambda+\mu)}$ and $E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu}$. It is known that the displacement method of linear elasticity in combination with low order conforming finite elements is not suitable for almost incompressible materials. These are materials where the Poisson ratio ν approaches $1/2$, or, in terms of the Lamé constants, where $\lambda \gg \mu$. This failure is called Poisson locking. One approach to avoid the locking effect is based on a mixed formulation, cf. Brezzi and Fortin [2] for a more detailed discussion. We consider

$$(1) \quad \begin{aligned} -2\mu \int_{\Omega} \varepsilon(u) : \varepsilon(v) dx &+ \int_{\Omega} \operatorname{div} v p dx = \int_{\Omega} f_0 v dx + \int_{\Gamma_1} f_1 v ds \\ \int_{\Omega} \operatorname{div} u q dx &- \frac{1}{\lambda} \int_{\Omega} p q dx = 0 \end{aligned}$$

$\forall v \in V, \forall q \in M$, where $V := \{v \in (H^1(\Omega))^d : v = 0 \text{ on } \Gamma_0\}$ and $M := L_2(\Omega)$. Here, u denotes the displacement vector and p the Lagrange multiplier or pressure.

Let us now consider a formal framework for saddle point problems with a penalty term. Let V and M be two Hilbert spaces with inner products $(\cdot, \cdot)_V, (\cdot, \cdot)_M$ and denote by $\|\cdot\|_V, \|\cdot\|_M$ the induced norms. Furthermore, let $a(\cdot, \cdot) : V \times V \rightarrow \mathbf{R}$, $b(\cdot, \cdot) : V \times M \rightarrow \mathbf{R}$, and $c(\cdot, \cdot) : M \times M \rightarrow \mathbf{R}$ be bilinear forms. Then, we consider the abstract problem

Find $(u, p) \in V \times M$, s.t.

$$(2) \quad \begin{aligned} a(u, v) + b(v, p) &= \langle F, v \rangle \quad \forall v \in V \\ b(u, q) - t^2 c(p, q) &= \langle G, q \rangle \quad \forall q \in M, \quad t \in [0, 1], \end{aligned}$$

where $F \in V'$ and $G \in M'$.

Problem (2) is well-posed under some assumptions on the bilinear forms. Let $a(\cdot, \cdot)$ be a continuous, symmetric, and V -elliptic bilinear form, i.e. $\exists \alpha > 0$, s.t. $a(v, v) \geq \alpha \|v\|_V^2 \quad \forall v \in V$, let $b(\cdot, \cdot)$ be a continuous bilinear form satisfying an inf-sup condition, i.e. $\exists \beta_0 > 0$, s.t. $\inf_{q \in M} \sup_{v \in V} \frac{b(v, q)}{\|v\|_V \|q\|_M} \geq \beta_0$, and let $c(\cdot, \cdot)$ be a continuous, symmetric and positive semi-definite bilinear form. Under these assumptions, the operator associated with Problem (2) is uniformly bounded with respect to the penalty term t . Note that these are not the most general assumptions for (2) to be uniquely solvable. For a proof and further discussions, see e.g. Braess [1]. This result also holds for suitable mixed finite elements, cf. [1]. The mixed formulation of linear elasticity clearly satisfies the assumptions made in the abstract formulation. Thus, (1) is a well-posed problem.

We discretize the saddle point problem (1) by a variant of the Taylor-Hood element, i.e. the $P_1(h) - P_1(2h)$ element. This element uses continuous, piecewise linear functions on a triangular mesh τ_h with the typical meshsize h for the displacement u and continuous, linear functions on a triangular mesh τ_{2h} with the meshsize $2h$ for the pressure p . Thus, the finite element spaces $V^h \subset V$ and $M^h \subset M$ are given by $V^h := \{v_h \in (\mathcal{C}(\Omega))^d \cap V : v_h \in \mathcal{P}_1 \text{ on } T \in \tau_h\}$ and

$M^h := \{q_h \in \mathcal{C}(\Omega) \cap M : q_h \in \mathcal{P}_1 \text{ on } T \in \tau_{2h}\}$. This results in a stable finite element method, cf. Brezzi and Fortin [2]. Discretizing (2) by $P_1(h) - P_1(2h)$ elements, we obtain a linear system of algebraic equations

$$(3) \quad \mathcal{A}x = \mathcal{F},$$

where

$$\mathcal{A} := \begin{pmatrix} A & B^t \\ B & -t^2 C \end{pmatrix}, \quad \mathcal{F} := \begin{pmatrix} F \\ G \end{pmatrix}.$$

3. Additive domain decomposition algorithms

We discuss three different additive domain decomposition methods. In order to keep our presentation simple, we consider for the rest of the paper the discrete problem using matrices and vectors instead of operators and functions. For simplicity, we also always have in mind the concrete problem of mixed linear elasticity.

Let τ_H be a coarse finite element triangulation of Ω into N subdomains Ω_i with the characteristic diameter H . By refinement of τ_H , we obtain a fine triangulation τ_h with the typical mesh size h . We denote by H/h the size of a subdomain without overlap. From the given overlapping domain decomposition $\{\Omega_i\}_{i=1}^N$, we construct an overlapping partition of Ω by extension of each Ω_i to a larger subregion Ω'_i consisting of all elements of τ_h within a distance $\delta > 0$ from Ω_i .

An important ingredient for the construction of our preconditioners are restriction matrices $R_i, i = 1, \dots, N$ which, applied to a vector of the global space, return the degrees of freedom associated with the interior of Ω'_i . For the description of the coarse part, we need an extra restriction matrix R_0^t constructed by interpolation from the degrees of freedom of the coarse to the fine triangulation. With respect to the partition of our saddle point problem into displacement and Lagrange multiplier (or hydrostatic pressure) variables, we can always assume an associated partition of our restriction matrices, i.e. $R_i = (R_{i,u} R_{i,p}), i = 0, \dots, N$.

3.1. An exact overlapping additive Schwarz method. Our first method can be formulated as an additive Schwarz method in the general Schwarz framework now well-known for the construction of preconditioners for symmetric positive definite problems, cf. Smith, Bjørstad, and Gropp [9]. It has the form

$$(4) \quad \mathcal{B}^{-1} = R_0^t \mathcal{A}_0^{-1} R_0 + \sum_{i=1}^N R_i^t \mathcal{A}_i^{-1} R_i,$$

where the \mathcal{A}_i are local problems associated with the subdomains and \mathcal{A}_0 is the coarse problem stemming from the coarse triangulation, cf. also Klawonn and Pavarino [6]. Schwarz methods can also be defined in terms of a space decomposition of $V^h \times M^h$ into a sum of local subspaces and a coarse space

$$V^h \times M^h = V_0^h \times M_0^h + \sum_{i=1}^N V_i^h \times M_i^h.$$

For the $P_1(h) - P_1(2h)$ finite elements, we define local problems with zero Dirichlet boundary conditions for both displacement and pressure variables on the internal subdomain boundaries $\partial\Omega'_i \setminus \partial\Omega$. Additionally, we impose zero mean value for the

pressure on each Ω'_i . Then, we obtain the subspaces

$$\begin{aligned} V_i^h &:= V^h \cap \left(H_0^1(\Omega'_i) \right)^d \\ M_i^h &:= \{ q_h \in M^h \cap L_0^2(\Omega'_i) : q_h = 0 \text{ on } \Omega \setminus \Omega'_i \}. \end{aligned}$$

Since we use different mesh sizes for the displacement and pressure triangulation, we have a minimal overlap of one pressure node, i.e. $\delta = 2h$. Using the restriction matrices R_i , our local problems \mathcal{A}_i are defined by $\mathcal{A}_i := R_i \mathcal{A} R_i^t$. The coarse problem $\mathcal{A}_0 := \mathcal{A}_H$ is associated with $V_0^h := V^H$, $M_0^h := M^H$ and R_0^t is the usual piecewise bilinear interpolation matrix between coarse and fine degrees of freedom.

Note that \mathcal{B}^{-1} is an indefinite preconditioner, and that it is well-defined since \mathcal{A}_0 and \mathcal{A}_i are regular matrices by construction. We are currently working on a theoretical analysis of this method.

3.2. A block-diagonal preconditioner. Our second method is given by

$$(5) \quad \mathcal{B}_D^{-1} := R_0^t \mathcal{D}_0^{-1} R_0 + \sum_{i=1}^N R_i^t \mathcal{D}_i^{-1} R_i,$$

where

$$\mathcal{D} := \begin{pmatrix} A & O \\ O & M_p \end{pmatrix},$$

and M_p denotes the pressure mass matrix associated with the fine triangulation τ_h . Here, we define our restriction matrices R_i , $i = 1, \dots, N$, s.t. the local problems are defined with zero Dirichlet boundary conditions for both displacement and pressure variables. In this case, we do not need the local mean value of the pressure to be zero since we do not have to solve local saddle point problems in this preconditioner. Analogous to the first preconditioner, we define the local problems as $\mathcal{D}_i := R_i \mathcal{D} R_i^t$ and the coarse problem $\mathcal{D}_0 := \mathcal{D}_H$. This approach can be interpreted as an inexact additive Schwarz method, where the exact subdomain solves are replaced by appropriate matrices. It can also be written as a block-diagonal preconditioner

$$\mathcal{B}_D^{-1} = \begin{pmatrix} \hat{A}^{-1} & O \\ O & \hat{M}_p^{-1} \end{pmatrix},$$

with $\hat{A}^{-1} := R_{0,u}^t A_0^{-1} R_{0,u} + \sum_{i=1}^N R_{i,u}^t A_i^{-1} R_{i,u}$ and $\hat{M}_p^{-1} := R_{0,p}^t M_{0,p}^{-1} R_{0,p} + \sum_{i=1}^N R_{i,p}^t M_{i,p}^{-1} R_{i,p}$. Analogous to the first preconditioner, we define the coarse and local problems as $A_0 := A_H$, $M_{0,p} := M_{p,H}$ and $A_i := R_{i,u} A R_{i,u}^t$, $M_{i,p} := R_{i,p} M_p R_{i,p}^t$. The spaces V_i^h , $i = 1, \dots, N$, V_0^h , and M_0^h are as in the previous subsection. Only the local spaces for the pressure are now of the form $M_i^h := \{ q_h \in M^h : q_h = 0 \text{ on } \Omega \setminus \Omega'_i \}$. Note that this preconditioner is symmetric positive definite and can be used with the preconditioned conjugate residual method (PCR). It can be analyzed in the framework of block-diagonal preconditioners for saddle point problems with a penalty term, cf. Klawonn [5].

3.3. A block-triangular preconditioner. Our third preconditioner is of block-triangular form where the block solvers are constructed by using an overlapping additive Schwarz method. This method cannot be directly formulated in the

Schwarz terminology. The preconditioner has the form

$$(6) \quad \mathcal{B}_T^{-1} := \begin{pmatrix} \hat{A} & O \\ B & -\hat{M}_p \end{pmatrix}^{-1},$$

where \hat{A} and \hat{M}_p are defined as in Section 3.2. This preconditioner is indefinite and can be analyzed as a block-triangular preconditioner for saddle point problems with a penalty term, cf. Klawonn [4] or Klawonn and Starke [7].

4. Numerical experiments

We apply our preconditioners to the problem of planar, linear elasticity, cf. Section 2. Without loss of generality, we use $E = 1$ as the value of Young's modulus. As domain, we consider the unit square, i.e. $\Omega = (0, 1)^2$ and we use homogeneous Dirichlet boundary conditions for the displacements on the whole boundary. In this case, our problem reduces to

$$(7) \quad \begin{aligned} -2\mu \int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} \operatorname{div} v \, p \, dx &= \int_{\Omega} f_0 \, v \, dx \quad \forall v \in (H_0^1(\Omega))^2 \\ \int_{\Omega} \operatorname{div} u \, q \, dx - \frac{1}{\lambda+\mu} \int_{\Omega} p \, q \, dx &= 0 \quad \forall q \in L_{2,0}(\Omega), \end{aligned}$$

cf. Brezzi and Fortin [2], Section VI.1.

We discretize this problem with the $P_1(h) - P_1(2h)$ elements from Section 2. The domain decomposition of Ω is constructed by dividing the unit square into smaller squares with the side length H . All computations were carried out using MATLAB. As Krylov space methods, we consider GMRES in combination with the preconditioners \mathcal{B}^{-1} and \mathcal{B}_T^{-1} , and we use the preconditioned conjugate residual method (PCR) with the preconditioner \mathcal{B}_D^{-1} . The initial guess is always zero and as a stopping criterion, we use $\|r_k\|_2/\|r_0\|_2 \leq 10^{-6}$, where r_k is the k -th residual of the respective iterative method. In all of our experiments, f is a uniformly distributed random vector and we use the minimal overlap $\delta = 2h$.

To see if our domain decomposition methods are scalable, we carried out some experiments with constant subdomain size $H/h = 8$, appropriately refined mesh size h , and increased number of subdomains $N = 1/H^2$. Our experiments indicate that all three domain decomposition methods give a scaled speedup, cf. Figures 1, 2, i.e. the number of iterations seems to be bounded independently of h and N . In Figure 3, we show a comparison of the iteration numbers of the different preconditioners for the incompressible limit case. Since we are using two different Krylov space methods, in order to have a unified stopping criterion, we ran the experiments presented in Figure 3 using the reduction of the relative error $\|e_k\|_2/\|e_0\|_2 \leq 10^{-6}$ as a stopping criterion. Here, e_k is computed by comparing the k -th iterate with the solution obtained by Gaussian elimination. To have a comparison between our domain decomposition methods and the best possible block-diagonal and block-triangular preconditioners (based on inexact solvers for A and M_p), we also include a set of experiments with these preconditioners using exact block-solvers for A and M_p , cf. Figure 3. From these results, we see that our exact additive Schwarz preconditioner has a convergence rate which is comparable to the one of the exact block-triangular preconditioner.

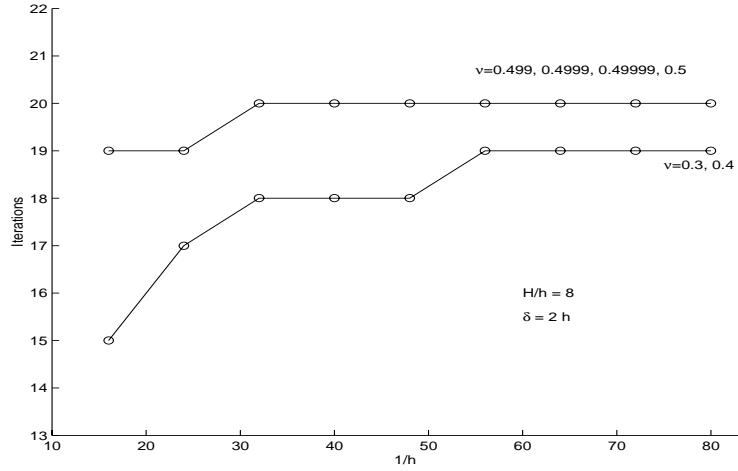


FIGURE 1. Elasticity problem with $P_1(h) - P_1(2h)$ finite elements: iteration counts for GMRES with overlapping additive Schwarz preconditioner \mathcal{B}^{-1}

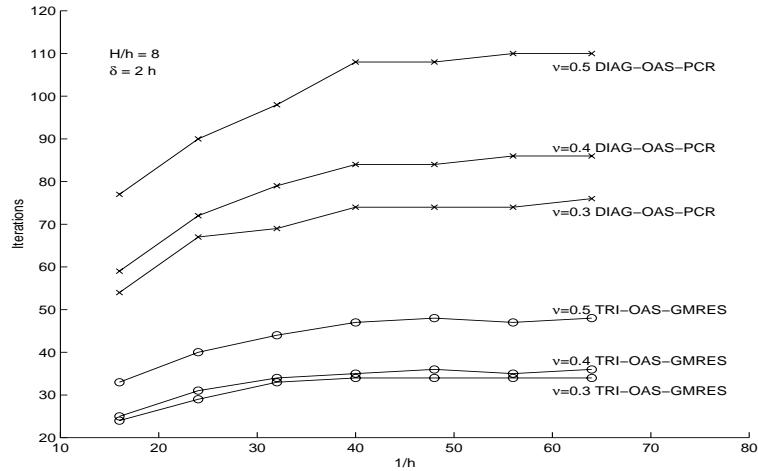


FIGURE 2. Elasticity problem with $P_1(h) - P_1(2h)$ finite elements: iteration counts for PCR with block-diagonal (DIAG-OAS-PCR) and GMRES with block-triangular (TRI-OAS-GMRES) preconditioners using the overlapping additive Schwarz preconditioner as block solver.

5. Conclusions

From our numerical results we see that the overlapping additive Schwarz approach is also a powerful way to construct preconditioners for saddle point problems. There is strong indication that scalability which is known to hold for symmetric positive definite problems also carries over to saddle point problems. Moreover, the exact overlapping additive Schwarz method gives results that are comparable to

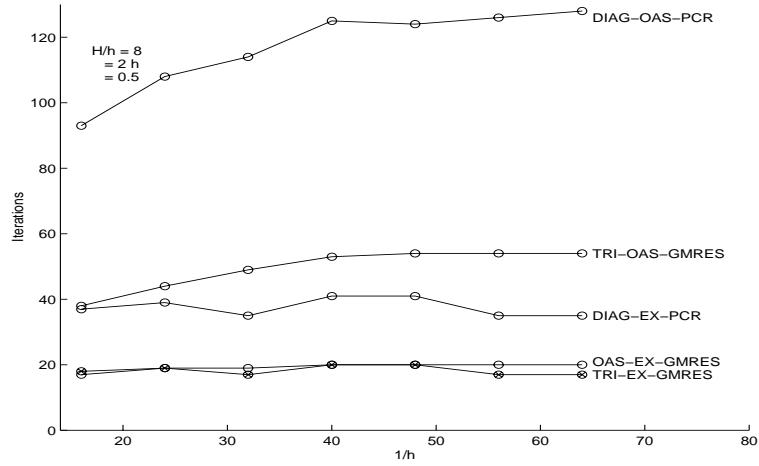


FIGURE 3. Elasticity problem with $P_1(h) - P_1(2h)$ finite elements: iteration counts for different preconditioners; OAS-EX-GMRES=GMRES with overlapping additive Schwarz and exact subdomain solvers, DIAG-OAS-PCR=PCR with block-diagonal preconditioner and overlapping additive Schwarz as block solvers, TRI-OAS-GMRES=GMRES with block-triangular preconditioner and overlapping additive Schwarz as subdomain solvers, DIAG-EX-PCR=PCR with block-diagonal preconditioner and exact block solvers, and TRI-EX-GMRES=GMRES with block-triangular preconditioner and exact block solvers.

those obtained under best of circumstances by the block-triangular preconditioner. The convergence rates of the exact additive Schwarz method are significantly faster than those obtained by the domain decomposition methods based on the block-diagonal and block-triangular approaches. We are currently working with Luca F. Pavarino on a more detailed comparison, taking into account also the complexity of the different preconditioners.

References

1. Dietrich Braess, *Finite elemente*, 2nd ed., Springer-Verlag, 1997.
2. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, Berlin, 1991.
3. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), no. 3, 604–620.
4. Axel Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comp. **19** (1998), no. 1, 172–184.
5. ———, *An optimal preconditioner for a class of saddle point problems with a penalty term*, SIAM J. Sci. Comp. **19** (1998), no. 2.
6. Axel Klawonn and Luca F. Pavarino, *Overlapping Schwarz methods for mixed linear elasticity and Stokes problems*, Tech. Report 15/97-N, Westfälische Wilhelms-Universität Münster, Germany, November 1997, Comput. Meth. Appl. Mech. Engrg., To appear.
7. Axel Klawonn and Gerhard Starke, *Block Triangular Preconditioners for Nonsymmetric Saddle Point Problems: Field-of-Values Analysis*, Tech. Report 04/97-N, Westfälische Wilhelms-Universität Münster, Germany, March 1997, Numer. Math., To appear.

8. Luca F. Pavarino, *Preconditioned mixed spectral element methods for elasticity and Stokes problems*, SIAM J. Sci. Comp. (1998), To appear.
9. B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.

INSTITUT FÜR NUMERISCHE UND INSTRUMENTELLE MATHEMATIK, WESTFÄLISCHE WILHEMS-UNIVERSITÄT, EINSTEINSTR. 62, 48149 MÜNSTER, GERMANY

E-mail address: klawonn@math.uni-muenster.de

Non-conforming Domain Decomposition Method for Plate and Shell Problems

Catherine Lacour

1. Introduction

The mortar element method is an optimal domain decomposition method for the approximation of partial differential equations on non-matching grids. There already exists applications of the mortar method to Navier-Stokes, elasticity, and Maxwell problems. The aim of this paper is to provide an extension of the mortar method to plates and shells problems. We first recall the Discrete Kirchhoff Triangles element method (D.K.T.) to approximate the plate and shell equations. The aim of this paper is then to explain what has to be changed in the definition of the D.K.T. method when the triangulation is nonconforming. Numerical results will illustrate the optimality of the mortar element method extended to shell problems and the efficiency of the FETI solution algorithm.

2. Recalling the D.K.T. method

We recall that the Koiter equations are deduced from the Naghdi equations (whose unknowns are the displacement of the mean surface $\vec{u} = (u_1, u_2, w)$ and the rotations $\underline{\beta} = (\underline{\beta}_1, \underline{\beta}_2)$ in the plane tangential to Ω) by imposing the Kirchhoff-Love relations, [1], given in (1), between the rotations $\underline{\beta}$ and the components of the displacement

$$(1) \quad \begin{aligned} \underline{\beta}_\alpha + w_{,\alpha} + b_\alpha^\lambda u_\lambda &= 0, \quad \text{where} \\ b_\alpha^\lambda &= a^{\lambda\mu} b_{\alpha\mu} \end{aligned}$$

and where we denote by $a_{\lambda\mu}$ the first fundamental form, by $b_{\alpha\mu}$ the second fundamental form and by $c_{\alpha\beta}$ the third fundamental form of the mean plane. The covariant derivatives are represented by a vertical bar and the usual derivatives by a comma. We use Greek letters for the indices in $\{1, 2\}$ and Latin letters for the indices in $\{1, 2, 3\}$.

1991 *Mathematics Subject Classification*. Primary 65F30; Secondary 65M60, 65Y05.
This paper is based on the PhD thesis [5].

2.1. Formulation of the problem.

- We consider a shell which is
- clamped along $\Gamma_0 \subset \Gamma = \partial\Omega$
 - loaded by a body force \vec{p}
 - loaded by a surface force applied to the part $\Gamma_1 = \Gamma - \Gamma_0 \times] - \frac{e}{2}; \frac{e}{2} [$ of its lateral surface, where e is the thickness.

We shall consider the following problem.

Find $(\vec{u}, \underline{\beta}) \in \vec{Z}$ such that

$$(2) \quad a[(\vec{u}, \underline{\beta}); (\vec{v}, \underline{\delta})] = l(\vec{v}, \underline{\delta}) \quad \forall (\vec{v}, \underline{\delta}) \in \vec{Z}$$

with

$$\begin{aligned} a[(\vec{u}, \underline{\beta}); (\vec{v}, \underline{\delta})] &= \int_{\Omega} e E^{\alpha\beta\lambda\mu} [\gamma_{\alpha\beta}(\vec{u}) \gamma_{\lambda\mu}(\vec{v}) + \frac{e^2}{12} \chi_{\alpha\beta}(\vec{u}, \underline{\beta}) \chi_{\lambda\mu}(\vec{v}, \underline{\delta})] \sqrt{a} ds^1 ds^2 \\ l(\vec{v}, \underline{\delta}) &= \int_{\Omega} \vec{p} \cdot \vec{v} \sqrt{a} ds^1 ds^2 + \int_{\Gamma_1} (\vec{N} \vec{v} - M^{\alpha} \underline{\delta}_{\alpha}) d\gamma \\ E^{\alpha\beta\lambda\mu} &= \frac{E}{2(1+\nu)} (a^{\alpha\lambda} a^{\beta\mu} + a^{\alpha\mu} a^{\beta\lambda} + \frac{2\nu}{1-\nu} a^{\alpha\beta} a^{\lambda\mu}) \\ \gamma_{\alpha\beta}(\vec{v}) &= \frac{1}{2} (v_{\alpha|\beta} + v_{\beta|\alpha}) - b_{\alpha\beta} w \\ \chi_{\alpha\beta}(\vec{v}, \underline{\delta}) &= \frac{1}{2} (\underline{\delta}_{\alpha|\beta} + \underline{\delta}_{\beta|\alpha}) - \frac{1}{2} (b_{\alpha}^{\lambda} v_{\lambda|\beta} + b_{\beta}^{\lambda} v_{\lambda|\alpha}) + c_{\alpha\beta} w \\ \sqrt{a} &= \sqrt{\det(a_{\alpha\beta})} \end{aligned}$$

and \vec{N} the resulting force on Γ_1 , $M = \epsilon_{\alpha\beta} M^{\beta} \vec{a}^{\alpha}$ the resulting moment on Γ_1 and \vec{Z} the space of the displacements/rotations which satisfy the Kirchhoff constraints and the boundary conditions.

The Discrete Kirchhoff Triangle method consists in defining a space of approximation \vec{Z}_h given by

$$\begin{aligned} \vec{Z}_h &= \{(\vec{v}_h, \underline{\delta}_h); v_{h\alpha} \in V_{h1}^k, \underline{\delta}_{h\alpha} \in V_{h1}^k, v_{h\alpha|_{\Gamma_0}} = 0, \quad \alpha = 1, 2; \\ &\quad w_h \in V_{h2}^k; w_{h|_{\Gamma_0}} = 0 \quad \forall T \in T_h; \underline{\delta}_h \text{ clamped} \\ (3) \quad (\vec{v}_h, \underline{\delta}_h)_T &\quad \text{satisfy the discrete Kirchhoff constraints given in [1]} \} \end{aligned}$$

such that V_{h1}^k is the space of P_2 -Lagrange elements and V_{h2}^k the space of P_3' -Hermite elements, h standing for a discretization parameter.

REMARK 1. The Kirchhoff relations are not satisfied at all the points in Ω and therefore the discrete space \vec{Z}_h is not included in \vec{Z} . This means that we have a non-conforming approximation of the Koiter equations.

3. The mortar element method for the D.K.T. approximation

Our purpose is to explain what has to be changed in the definition of the D.K.T. method when the triangulation is nonconforming. We recall that in order to match interface fields, we associate to the nonoverlapping decomposition of the domain Ω the skeleton of the decomposition and we choose the mortar and nonmortar sides [3].

For the shell equation, many functions have to be matched. First, we have the tangential displacements $v_{h\alpha}$ and then the transversal displacement w_h . We have to match also the rotations $\underline{\beta}_h$ associated with the displacement. For the first two components of the displacement, the matching is easy since these functions are independent and are involved in a second order equation. Their natural space is $H^1(\Omega)$ and the standard mortar method for piecewise parabolic elements is used. We recall that it involves the space of traces W_{h1} of functions of V_{h1}^k on the nonmortar sides and the subspace \tilde{W}_{h1} of W_{h1} of functions that are linear on the first and last (1D) element of the triangulation of this nonmortar side.

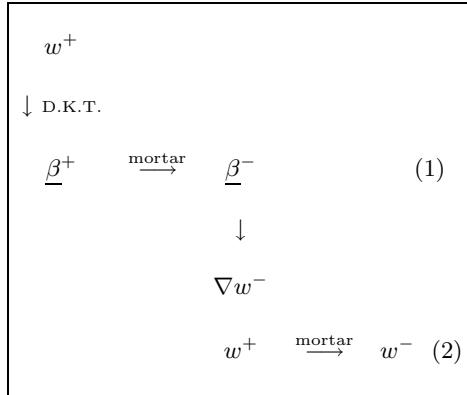
Let us state the matching across one particular non-mortar γ^* and denote by $+$ the mortar (master) side and by $-$ the nonmortar (slave) side of the decomposition. Then, for any function $v_{h\alpha}$, $\alpha = 1, 2$, we impose

$$(4) \quad \forall \psi_h \in \tilde{W}_{h1}, \quad \int_{\gamma^*} (v_{h\alpha}^- - v_{h\alpha}^+) \psi_h \, d\tau = 0.$$

The space V_{h1} of approximation for the global tangential components of the displacements is thus given by

$$(5) \quad V_{h1} = \{v_h \in L^2(\Omega), \quad v_h|_{\Omega^k} \in V_{h1}^k \text{ and} \\ \text{satisfies (4) along any non-mortar } \gamma^*\}.$$

The originality in the matching presented in this paper lies in the treatment of the out of plane displacement and the associated rotations. We recall that the D.K.T. condition is a relation between the displacements and the rotations, see formula (1). The nonmortar side values are recovered from the mortar side in the following two steps.



We start from w^+ given on the master side of the mortar, and then obtain $\underline{\beta}^+$ by using the D.K.T. condition.

Step (1)

We match $\underline{\beta}^-$ and $\underline{\beta}^+$ by the mortar relations. These are different for the normal and the tangential components (normal and tangential with respect to the interface). First, we match the tangential rotation $\underline{\beta}^{t-}$ by defining a relation between two piecewise second order polynomials. The relation is naturally the same as for the displacements $v_{h\alpha}$, $\alpha = 1, 2$. We then impose

$$(6) \quad \forall \psi_h \in \tilde{W}_{h1}, \quad \int_{\gamma^*} (\underline{\beta}^{t-} - \underline{\beta}^{t+}) \psi_h \, d\tau = 0.$$

Let us turn now to the normal rotations. We note first that from the D.K.T. conditions, the normal rotations are piecewise linear on the mortar side, [2]. Since we want to preserve, as much as possible, the Kirchhoff conditions, we shall glue the normal rotations as piecewise linear finite element functions. To do this, we define W_{h0} as being the set of continuous piecewise linear functions on γ^* (provided with the nonmortar triangulation) and \tilde{W}_{h0} as the subset of those functions of W_{h0} that are constant on the first and last segment of the (nonmortar) triangulation. We then impose the following relation between the (piecewise linear) normal rotations.

$$(7) \quad \forall \psi_h \in \tilde{W}_{h0}, \quad \int_{\gamma^*} (\underline{\beta}^{n-} - \underline{\beta}^{n+}) \psi_h \, d\tau = 0.$$

Step (2)

Now that the rotations are completely glued together and are uniquely defined over the interface from the corner values of $\underline{\beta}^-$ and all nodal values of $\underline{\beta}^+$ (themselves derived from v_α^+ , w^+), we specify the relations that define w^- . The first set of constraints is to satisfy “inverse D.K.T. conditions” i.e. to match the values of w^- with the rotations $\underline{\beta}^-$. We impose that

- the tangential derivatives of w coincide with $\underline{\beta}^{t-} + v_\alpha^-$ at any Lagrange node (vertex and middle point). This allows us to define a piecewise P_2 function on the nonmortar elements of γ^* and
- the normal derivative of w coincides with $\underline{\beta}^{n-}$ at each vertex of the triangulation of γ^* .

Since $\underline{\beta}^{n-}$ is piecewise linear, the D.K.T. condition is automatically satisfied at the middle node of each element. Furthermore, since $\underline{\beta}^{t-}$ is piecewise quadratic, it coincides with the tangential derivative of w^- not only at the nodal points but also on the whole interface γ^* . Since the tangential derivative of w^- is determined, it suffices to impose the value of w^- at one of the endpoints of γ^* to determine the value of w^- entirely. We impose, with $\gamma^* = [p_1, p_2]$,

$$(8) \quad \forall \psi \in \tilde{W}_{h0}, \quad \int_{\gamma^*} (w^- - w^+) \psi \, d\tau = 0$$

$$(9) \quad w^-(p_1) = w^+(p_1)$$

$$(10) \quad w^-(p_2) = w^+(p_2)$$

which can be seen as the relations that determine the nodal values of w^- that are lacking. We insist on the fact that this construction leads to finite element functions that satisfy the D.K.T. conditions on each interface. This allows us to define the global space V_{h2} of transversal displacements as follows.

$$(11) \quad V_{h2} = \{w_h \in L^2(\Omega) \mid w_{h|\Omega^k} \in V_{h2}^k \text{ and satisfy (6), (7), (9), (8) and (10)}\}$$

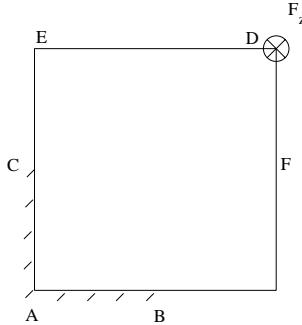


FIGURE 1. Plate configuration

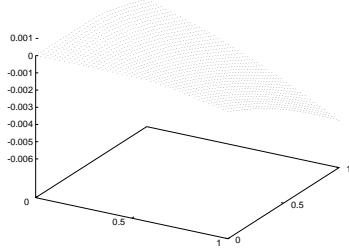


FIGURE 2. Deformation of the plate

4. Numerical results

Description of the problem.

We consider the plate, given in Figure 1, with the following properties.

Thickness : $e = 0.05 \text{ m}$

Length : $L = 1 \text{ m}$

Width : $w = 1 \text{ m}$

Properties : $E = 1.0E7 \text{ Pa}$ and $\nu = 0.25$

Boundary conditions : AB and AC clamped

Loading force : on D : $F_z = -1.0 \text{ N}$

1. Matching results.

The deformation of the plate loaded at the point D is shown in Figure 2.

The plate is now decomposed as in Figure 3 with non matching grids on the interfaces. The first results, given in Figures 4 and 5, show the good matching of the transversal displacement on the section CF and of the normal derivative of the transversal displacement.

2. Scalability results.

The discretization leads to an algebraic saddle-point problem that can be solved by the FETI method introduced in [4] and [6]. The FETI method presented here

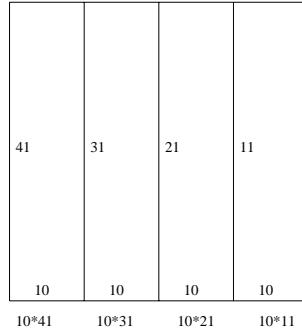
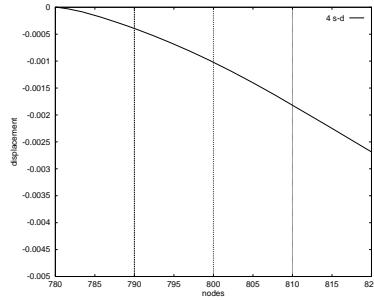
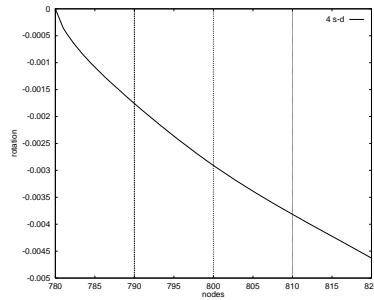


FIGURE 3. Example of decomposition and mesh

FIGURE 4. Transversal displacement on *CF*FIGURE 5. Normal derivative of the transversal displacement on *CF*

results in a scalable substructuring algorithm for solving this saddle point problem iteratively.

For this plate problem approximated by the D.K.T. finite element method, we observe that for a fixed local mesh, the number of iterations is independent of the number of subdomains. Thus, the parallel implementation exhibits good scalability when the right preconditioner is used [5], cf. Table 1.

TABLE 1. Scalability results

Number of sub domains	Iterations	Residual
4 (2 × 2)	49	7.10E-004
8 (4 × 2)	76	8.202E-004
16 (4 × 4)	77	7.835E-004
32 (8 × 4)	96	7.166E-004

Without preconditioner.

Number of sub domains	Iterations	Residual
4 (2 × 2)	14	5.256E-004
8 (4 × 2)	16	7.566E-004
16 (4 × 4)	16	8.966E-004
32 (8 × 4)	16	9.662E-004
64 (8 × 8)	16	8.662E-004

With preconditioner.

5. Conclusion

Analysis of the application of the D.K.T. method extended to nonconforming domain decomposition to shell problem illustrate the optimality of the mortar element method and the efficiency of the FETI solution algorithm.

References

1. M. Bernadou, *Méthodes d'éléments finis pour les problèmes de coques minces*, RMA, Masson, Springer Verlag co-publication, 1994.
2. M. Bernadou, P. Matao Eiroa, and P. Trouve, *On the convergence of a D.K.T. method valid for shells of arbitrary shape*, Rapport de Recherche INRIA, No 2010 **17** (1993).
3. C. Bernardi, Y. Maday, and A.T. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, Publications du laboratoire d'Analyse Numérique de Paris VI (1990).
4. C. Farhat and M. Geradin, *Using a reduced number of Lagrange multipliers for assembling parallel incomplete field finite element approximations*, Computer Methods in Applied Mechanics and Engineering (1992).
5. C. Lacour, *Analyse et résolution numérique de méthodes de sous-domaines non conformes pour des problèmes de plaques*, PhD dissertation, University Pierre et Marie Curie, 4, place Jussieu, Paris, January 1997.
6. F.X. Roux, *Méthode de décomposition de domaine à l'aide de multiplicateurs de Lagrange et application à la résolution en parallèle des équations de l'élasticité linéaire*, PhD dissertation, University Pierre et Marie Curie, 4, place Jussieu, Paris, December 1989.

LABORATOIRE D'ANALYSE NUMÉRIQUE DE PARIS VI, TOUR 55-65, 5EME ÉTAGE, 4 PLACE JUSSIEU, 75252 PARIS CEDEX 05, FRANCE

E-mail address: lacour@ann.jussieu.fr

Solutions of Boundary Element Equations by a Flexible Elimination Process

Choi-Hong Lai and Ke Chen

1. Introduction

Field methods such as finite difference, finite volume or finite element methods are usually applied to solve partial differential equations. Such methods reduce either linear partial differential equations or linearized partial differential equations to a large sparse set of linear equations. However for certain kinds of boundary value problems, boundary element methods have proven to be effective alternatives, especially when dealing with exterior problems. One well-known advantage of boundary element methods is that the dimension of the original problem is reduced by one. The reason is because a differential problem in a domain $\Omega \subset \mathbf{R}^m$ can be reformulated as an integral equation problem [2] over the underlying boundary $\Gamma = \partial\Omega \subset \mathbf{R}^{m-1}$. A number of approximations can then be applied that may lead to boundary element equations.

This paper has two objectives. Firstly, a brief description is given of the sequential boundary element method followed by a possible conversion and its requirements to a distributed algorithm. Secondly, a flexible elimination method [10] is used with the distributed algorithm to solve the set of boundary element equations. Such an elimination method has the advantage of not following the usual ordering of the system of equations in a classical elimination procedure such as Gaussian elimination. The technique greatly enhances the intrinsic parallelism of solving dense linear systems of equations. This paper also compares the accuracy of the flexible elimination method with the classical Gauss-Jordan elimination method for two potential flow problems and provides some timing results of the flexible elimination method on a network of SUN workstations using MPI as distribution directives. The paper concludes with an extension of the algorithm to complex systems of linear equations.

1991 *Mathematics Subject Classification*. Primary 65Y05; Secondary 65R20, 65F05.

Key words and phrases. Boundary Elements, Flexible Elimination, Parallel Algorithms, Distributed Computing.

This research is supported by a London Mathematical Society Scheme 4 Grant (Ref 4222).

2. Boundary Element Methods

For certain kinds of boundary value problems such as Laplace's equation, the Helmholtz equation and the linear diffusion equation, boundary element methods have proved to be effective alternatives to field methods. It is particularly true for exterior problems. One typical technique similar to the boundary element method known as the panel element method [5], which is a well established practice in aeronautical engineering industry for the design of steady and unsteady subsonic compressible flows over airfoils and other airframe structures, is proved to be a successful tool for engineers.

Consider the exterior Neumann problem in two dimensions using a simple layer logarithmic potential [6]. Let $\partial\Omega$ denote the surface of a body which is sufficiently smooth and Ω denote the exterior of the body where a harmonic function is defined in it. Suppose the outward normal derivative ϕ' along $\partial\Omega$ is known, then the solution can be written as

$$(1) \quad \phi(p) = \int_{\partial\Omega} \ln |p - q| \sigma(q) ds, \quad p \in \Omega \cup \partial\Omega$$

Suppose the source density σ is distributed on $\partial\Omega$, then it must satisfies the integral

$$(2) \quad \int_{\partial\Omega} \frac{\partial}{\partial n_i} \ln |q_i - q| \sigma(q) ds + \pi \sigma(q_i) = \phi'(q_i), \quad q_i \in \partial\Omega$$

where n_i is the outward normal at q_i . Suppose now the boundary $\partial\Omega$ is subdivided into elements $\partial\Omega_i$, $i = 1, \dots, n$, the above integral can be approximated by

$$(3) \quad \sum_{j=1}^n \sigma_j \int_{\partial\Omega_i} \frac{\partial}{\partial n_i} \ln |q_i - q_j| ds + \pi \sigma_i = \phi'_i$$

where $\sigma_j = \sigma(q_j)$ and $\phi'_i = \phi'(q_i)$. The discretized replacement results in the set of dense linear equations $A\underline{\sigma} = \underline{b}$ where $\underline{\sigma} = [\sigma_1 \dots \sigma_n]^\top$ and $\underline{b} = [\phi'_1 \dots \phi'_n]^\top$. It involves far fewer unknowns than any field method such as finite difference or finite volume methods. Hence a direct method such as Gaussian elimination is usually sufficient for moderate n .

For large n , a direct method can be expensive. Our work here will be a good starting point towards achieving speed up. Iterative methods are alternative approaches that have achieved a varied level of success. That is, efficient iterative solvers can be problem-dependent and preconditioners dependent; refer to [1].

2.1. Sequential Algorithm. It is clear that a sequential boundary element method involves two computational functionals, namely, (a) the construction of the dense matrix A and (b) the solution of $A\underline{\sigma} = \underline{b}$, which cannot be computed concurrently. However, one can easily parallelize functional (a) because of the intrinsic parallelism existing in (3). Early work in the parallelization of these integrals can be found in [3, 8] and the references therein. The parallelization essentially involves the sharing of the computation of the above integrals amongst a number of processors within a distributed environment. Then functional (b) may be started once functional (a) is completed. There are a large number of literatures on parallel solvers for dense matrices. However on some parallel machines Gauss-Jordan elimination is preferred. One common feature of these parallel implementations is that they primarily rely on the extraction of parallelism

to Gaussian or Gauss-Jordan elimination. The intrinsic sequential behaviour of the two functionals has not been removed to suit modern days distributed computing.

Suppose t_a and T_a denote the sequential and parallel times respectively for computing the integrals and t_s and T_s denote the sequential and parallel times respectively for solving $A\sigma = \underline{b}$, then the total sequential computing time, t , is given by

$$(4) \quad t = t_a + t_s$$

and the corresponding parallel computing time, t_p , is given by

$$(5) \quad t_p = T_a + T_s.$$

2.2. A Distributed Algorithm. Divided and conquer type algorithms are usually used to tackle discretized problems in distributed and/or parallel computing environments. There are notably three major classes of divide and conquer type of algorithms, namely, domain decomposition [7], problem partitioning [9] and functional decomposition [4]. The first two type of algorithms concern geometric partitioning of computational domains according to either load balancing or regional physical/numerical behaviour and the latter concerns parallelism in computational functionals. For the present problem, if functionals (a) and (b) could be performed concurrently, then it is possible to achieve

$$(6) \quad t_d = \max\{T_a, T_s\}$$

It is certainly true that $t_p > t_d$. Hence the key requirement of a new dense matrix solver is that it does not rely on the natural ordering of the equations as required by the classical Gaussian elimination. In other words, the matrix solver should be able to eliminate any equations that have been constructed by the parallel or distributed processing of functional (a) and are made available to the matrix solver in a random ordering. It is important that such distributed algorithm should not affect the accuracy of the solution compared to that obtained by means of the classical Gaussian elimination.

Now suppose the obstacle surface is subdivided into n_p sub-domains such that n/n_p is an integer for simplicity. It is also assumed that each sub-domain is mapped to a processor within the distributed computing environment. Then a distributed algorithm can be given as follow.

Algorithm: A distributed boundary element method.

Notation:- n (number of elements), $\partial\Omega$ (shape of the body),

n_p (number of processors and n/n_p is an integer for simplicity),

i_r (maps the local element numbering r to

the global element numbering),

g_i (denotes the arrival ordering of the equations at sub-task 2).

```

sub-task 1 {
    parallel-for  $p = 1, \dots, n_p$ 
        for  $r = 1, \dots, n/n_p$ 
            Compute row  $i := i_r$  of matrix  $A$ ;
            Compute element  $[\underline{b}]_i$  of the r.h.s. vector  $\underline{b}$ ;
            Non-blocking send of row  $i$  and element  $[\underline{b}]_i$ ;
        end for
    end parallel-for
}

```

```

}end sub-task 1

sub-task 2 {
  for  $i = 1, \dots, n$ 
    Block receive row  $g_i$  and  $[\underline{b}]_{g_i}$  from sub-task 1;
    Flexible elimination step (see next Section);
  end for
}end sub-task 2

end Algorithm

```

3. A Flexible Elimination Method

The method is based on the concept of orthogonality of vectors. Suppose the coefficients of the i th equation of the system $A\underline{\sigma} = \underline{b}$, where $[A]_{ij} = a_{ij}$, $[\underline{b}]_i = b_i$ and $[\underline{\sigma}]_i = x_i$, are written as the augmented vector $A_i = [a_{i1}a_{i2}\cdots a_{in} - b_i]^\top$, $i = 1, 2, \dots, n$, then a vector V is said to be the solution of the system provided that the last component of V is unity and that $A_i^\top V = 0$, for all $i = 1, 2, \dots, n$. Let $\mathcal{C} = \{A_1, A_2, \dots, A_n\}$. Define the set $\mathbf{C}^{(i)} = \{C_1, C_2, \dots, C_i \mid \text{a selection of } i \text{ different vectors from } \mathcal{C}\}$ such that $\mathbf{C}^{(i)} = \mathbf{C}^{(i-1)} \cup \{C_i\}$ and the set $\mathbf{R}^{(i)}$ as the subspace of dimension $n+1-i$ which consists of vectors orthogonal to the vectors in $\mathbf{C}^{(i)}$, for $i = 1, 2, \dots, n$. It is intuitively obvious that the basis for the $(n+1)$ -dimensional subspace $\mathbf{R}^{(0)}$ may be chosen as the natural basis, i.e.

$$(7) \quad \mathbf{V}^{(0)} = \{V_1^{(0)} := [10 \cdots 0]^\top, \dots, V_{n+1}^{(0)} := [0 \cdots 01]\}$$

For each i from 1 to n , linear combinations of a chosen vector from the basis $\mathbf{V}^{(i-1)}$ and one of the remaining vectors from that basis are performed. Such linear combinations are subject to the condition that the resulting vectors are orthogonal to C_i . Therefore for any $C_i \in \mathcal{C} \setminus \mathbf{C}^{(i-1)}$, it is equivalent to the construction of the basis

$$(8) \quad \mathbf{V}^{(i)} = \left\{ V_k^{(i)} \in \mathbf{R}^{(i)} \mid V_k^{(i)} := \alpha_k V_{s(k)}^{(i-1)} + V_{m(k)}^{(i-1)}, C_i^\top V_k^{(i)} = 0 \right\}$$

where $1 \leq k \in \mathbf{N} \leq n+1-i$, $s(k)$ and $m(k) \in \mathbf{N}$ and $s(k) \neq m(k)$. Here $\mathbf{C}^{(0)} = \{\emptyset\}$ is empty. It can be easily shown that

$$(9) \quad \alpha_k = -\frac{C_i^\top V_{m(k)}^{(i-1)}}{C_i^\top V_{s(k)}^{(i-1)}}$$

and that the vector $V_k^{(i)}$ is orthogonal to each of the vectors in $\mathbf{C}^{(i)} \subset \mathcal{C}$. In order to avoid instability of the orthogonalization procedure, the condition $C_i^\top V_{s(k)}^{(i-1)} \neq 0$ must be satisfied. Usually a check may be incorporated in the algorithm to ensure the stability of the method. The dimension of the subspace $\mathbf{R}^{(n)}$ is 1 and the basis $\mathbf{V}^{(n)}$ is orthogonal to every vector in $\mathbf{C}^{(n)} = \mathcal{C}$. Thus the solution of the system $A\underline{\sigma} = \underline{b}$ is constructed [10]. It should be noted here that when C_i is chosen as A_i and that if $s(k) = 1$ and $m(k) = k+1$ then the method is equivalent to a Gauss-Jordan elimination [10].

However the choice of $s(k)$ and $m(k)$ can be as flexible as it could be, provided that the condition $s(k) \neq m(k)$ is satisfied. From (8), $n+1-i$ pairs of vectors are

chosen from the basis of $\mathbf{V}^{(i-1)}$, such that no two pairs of such vectors are identical, in order to perform the linear combinations. Note that the linear combinations are performed by using the constant α_k , as given by (9), which involves the division of two floating point numbers. Therefore α_k will loose accuracy if the two floating point numbers are of very different orders of magnitudes. One criterion which governs the choice of $s(k)$ and $m(k)$ is to ensure similar order of magnitude of the floating point numbers $C_i^\top V_{m(k)}^{(i-1)}$ and $C_i^\top V_{s(k)}^{(i-1)}$. This involves some additional logical comparison work. It is possible to include tests in an implementation to check that either pivoting is needed or redundant equations have occurred. It can easily be seen that the data structure of the solution vector, i.e. $V_1^{(n)}$, is not affected with various choices of $s(k)$ and $m(k)$. It should be mentioned here that pivoting is equivalent to suitable choices of $s(k)$ and $m(k)$. Therefore the implication is that column pivoting strategy has no effect on the data structure of the solution vector and that the same property applies to row pivoting strategy as long as $s(k)$ is not the same as $m(k)$. More details of these properties and examples can be found in [10].

As far as distributed computing is concerned, the flexible choice of s and m is not the key ingredient. However, if we consider the choice of $\mathbf{C}^{(i)}$, we realize that there is no preference in the order of choosing vectors for $\mathbf{C}^{(i)}$ from \mathcal{C} . In fact the order of choosing C_i is not important in the present algorithm. The only two criteria governing the choice of C_i is (i) $\mathbf{C}^{(i)}$ consists of a selection of i different vectors from \mathcal{C} and (ii) eqn (9) must be satisfied in order to achieve stability of the algorithm. Hence it is possible to choose $A_{g(i)}$ provided that the map $g : \mathbf{N} \rightarrow \mathbf{N}$ is one-to-one and that $1 \leq g(i) \leq n$, $i = 1, \dots, n$ where $g(i) \neq g(j)$ if $i \neq j$. Such mapping of g implies the order of elimination process is not as rigid as that in a Gauss-Jordan elimination. At any step i , only $C_i \in \mathbf{C}^{(i)} \subset \mathcal{C}$ and $\mathbf{V}^{(i-1)}$ are required in the computation. Therefore the orthogonalization procedure can be completely separated from the knowledge of the set $\mathcal{C} \setminus \mathbf{C}^{(i)}$. In terms of functionals (a) and (b), the requirement for functional (b) to follow functional (a) in a sequential processing is removed. This particular property satisfies the concurrent processing of both functionals (a) and (b) as the necessary requirement described in the previous Section. In terms of the sub-tasks as described in the previous Section, sub-task 2 will be allowed to take and process any equation arriving at its door without jeopardizing the data structure and the stability of the elimination process.

One can also easily see that, by choosing $s(k) = 1$ and $m(k) = k + 1$, the algorithm is particularly suitable for vector calculation and the scalar products involved in the algorithm can be optimized to provided faster timings. We shall investigate the accuracy of the algorithm by using a random number generator to provide a re-ordering function $g(i)$.

4. Examples

For simplicity, potential flows past over obstacles at zero angle of attack are considered. The two obstacles under consideration are (i) an ellipse described by $x^2 + \frac{y^2}{4} = 1$ and (ii) the NACA0012 airfoil. It is assumed that the variables in (3) are normalized with respect to the far field velocity.

Test 1. The algorithm is first implemented as a Gauss-Jordan elimination method by taking $s(k) = 1$ and $m(k) = k + 1$ for the solution of the boundary

TABLE 1. Timings in seconds for solving boundary element equations.

n	$n_p = 1$	$n_p = 4$	$n_p = 8$
640	326	232	152.5
1024	1320	957	814.3

element equations with the natural ordering of the system. Having solved the system of equations for $\sigma(q)$, it is possible to evaluate $\phi(q)$ using (1) and hence the tangential velocity \mathbf{V} along the surface of the obstacle. Pressure coefficients $C_p = 1 - |\mathbf{V}|^2$ [5, 6] along the surface of the obstacle can be evaluated. Then a random number generator is used to provide a re-ordering function $g(i)$ as described above. The re-ordering function serves the same functionality as providing rows of matrix coefficients from sub-task 1 to sub-task 2 at a different ordering from the natural ordering according to the element numbering.

Pressure coefficients are obtained for test cases (i) and (ii) by a Gauss-Jordan elimination method using the natural ordering of the equations and the re-ordering of the equations. The maximum errors that have been recorded for both cases are less than 4 decimal places.

Test 2. The algorithm is also run on a network of Sun SPARC Classic workstations at Greenwich. MPI Standard was used to implement the communication. Timings for the solutions of 640 and 1024 unknowns were recorded for both of the obstacle configurations as a sequential process and distributed processes on 4 and 8 workstations. The re-ordering function described above is used here. Table 1 shows the timings on the network. However the speedup in this test is not good because of the heavily used network.

5. Extension to Complex Systems of Equations

Suppose the system $A\underline{\sigma} = \underline{b}$ is a complex system such that $A = A_1 + iA_2$, $\underline{\sigma} = \underline{\sigma}_1 + i\underline{\sigma}_2$ and $\underline{b} = \underline{b}_1 + i\underline{b}_2$. The complex system can be re-written as the following real system

$$\begin{bmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{bmatrix} \begin{bmatrix} \underline{\sigma}_1 \\ \underline{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \end{bmatrix}$$

Since the ordering of the system in the flexible elimination algorithm does not affect the solution, once the $g(i)$ th equation is constructed and is made available to sub-task 2, the two equations

$$\begin{bmatrix} [A_1]_{g(i)} & [-A_2]_{g(i)} & [-\underline{b}_1]_{g(i)} \\ [A_2]_{g(i)} & [A_1]_{g(i)} & [-\underline{b}_2]_{g(i)} \end{bmatrix}$$

can be used immediately into the construction of the new basis as described previously in (8).

6. Conclusions

A distributed algorithm for boundary element methods is discussed. In order to introduce concurrency to boundary element methods at the functional level, one has to employ a flexible elimination method as described in this paper. The accuracy of the flexible elimination method is good and its stability can be ensured easily. Early distributive computing tests show that the method is a suitable candidate for

solving boundary element equations in a distributive environment. The extension to a complex system of equation is straightforward.

References

1. K. Chen, *Preconditioning boundary element equations*, Boundary elements: implementation and analysis of advanced algorithms (W. Hackbusch & G. Wittum, ed.), no. 54, Vieweg, 1996.
2. D. Colton and R. Kress, *Integral equation methods in scattering theory*, Wiley, 1993.
3. A.J. Davies, *The boundary element method on the ICL DAP*, Parallel Computing **8** (1988), 348–353.
4. I. East, *Parallel processing with communicating process architecture*, UCL Press Ltd, London, 1995.
5. J.L. Hess and A.M.O. Smith, *Calculations of potential flow about arbitrary bodies*, Progress in Aeronautical Sciences (D. Kucheman, ed.), no. 8, 1976.
6. M.A. Jawson and G.T. Symm, *Integral equation methods in potential theory and elastostatics*, Academic Press, 1977.
7. D.E. Keyes, Y. Saad, and D.G. Truhlar, *Domain-based parallelism and problem decomposition methods in computational science and engineering*, SIAM, 1995.
8. C-H. Lai, *A parallel panel method for the solution of fluid flow past an aerofoil*, CONPAR88 (CR Jesshope and KD Reinartz, eds.), Cambridge University Press, 1989, pp. 711–718.
9. ———, *A domain decomposition for viscous/inviscid coupling*, Advances in Engineering Software **26** (1995), 151–159.
10. ———, *An extension of Purcell's vector method with applications to panel element equations*, Computers Math. Appl. **33** (1997), 101–114.

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH,
WELLINGTON STREET, WOOLWICH, LONDON SE18 6PF, UK

E-mail address: C.H.Lai@greenwich.ac.uk and <http://cms1.gre.ac.uk/>

DEPARTMENT OF MATHEMATICAL SCIENCES, UNIVERSITY OF LIVERPOOL, PEACH STREET,
LIVERPOOL L69 3BX, UK

E-mail address: k.chen@liverpool.ac.uk and <http://www.liv.ac.uk/mathematics/applied>

An Efficient FETI Implementation on Distributed Shared Memory Machines with Independent Numbers of Subdomains and Processors

Michel Lesoinne and Kendall Pierson

1. Introduction

Until now, many implementations of the FETI method have been designed either as sequential codes on a single CPU, or as parallel implementations with a *One Subdomain per Processor* approach. This approach has been particularly typical of implementations on distributed memory architectures such as the IBM SP2. In the last couple of years, several computer manufacturers have introduced new machines with a Distributed Shared Memory (DSM) programming model –e.g. SGI Origin 2000, or HP Exemplar. In such architectures, the physical memory is distributed among the processors or CPU boards but any memory location can be accessed logically by any CPU independently of where the particular memory page being accessed has physically been allocated. As more and more machines of this type are available with a relatively small number of processors, the interest in implementing FETI with an independent number of subdomains and processor has increased. We report on such an implementation of FETI and highlight the benefits of this feature. We have found that medium size to large problems can be solved even on a sequential machine with time and memory requirements that are one to two order of magnitude better than a direct solver.

2. Objectives

When writing our new FETI code, the main objectives were:

- Efficient data structures for distributed shared memory
- Number of subdomains independent of the number of processors

The second requirement was the most important requirement and, when taken to the extreme of a single processor, naturally leads to being able to run the same code sequentially.

1991 *Mathematics Subject Classification*. Primary 65Y05; Secondary 65N55, 65Y10.
The first author acknowledges partial support by ANSYS, Inc.

3. Overview of FETI

In order to keep this paper self-contained as much as possible, we begin with an overview of the original FETI method [6, 1, 2, 4, 5].

The problem to be solved is

$$(1) \quad Ku = F$$

where K is an $n \times n$ symmetric positive semi-definite sparse matrix arising from the finite element discretization of a second- or fourth-order elastostatic (or elastodynamic) problem defined over a region Ω , and F is a right hand side n -long vector representing some generalized forces. If Ω is partitioned into a set of N_s disconnected substructures $\Omega^{(s)}$, the FETI method consists in replacing Eq (1) with the equivalent system of substructure equations

$$(2) \quad \begin{aligned} K^{(s)} u^{(s)} &= F^{(s)} - B^{(s)T} \lambda & s = 1, \dots, N_s \\ \Delta &= \sum_{s=1}^{N_s} B^{(s)} u^{(s)} = 0 \end{aligned}$$

where $K^{(s)}$ and $F^{(s)}$ are the unassembled restrictions of K and F to substructure $\Omega^{(s)}$, λ is a vector of Lagrange multipliers introduced for enforcing the constraint $\Delta = 0$ on the substructure interface boundary $\Gamma_I^{(s)}$, and $B^{(s)}$ is a signed Boolean matrix that describes the interconnectivity of the substructures. A more elaborate derivation of (2) can be found in [6, 3]. In general, a mesh partition may contain $N_f \leq N_s$ floating substructures — that is, substructures without enough essential boundary conditions to prevent the substructure matrices $K^{(s)}$ from being singular — in which case N_f of the local Neumann problems

$$(3) \quad K^{(s)} u^{(s)} = F^{(s)} - B^{(s)T} \lambda \quad s = 1, \dots, N_f$$

are ill-posed. To guarantee the solvability of these problems, we require that

$$(4) \quad (F^{(s)} - B^{(s)T} \lambda) \perp \text{Ker}(K^{(s)}) \quad s = 1, \dots, N_f$$

and compute the solution of Eq. (3) as

$$(5) \quad u^{(s)} = K^{(s)+} (F^{(s)} - B^{(s)T} \lambda) + R^{(s)} \alpha^{(s)}$$

where $K^{(s)+}$ is a generalized inverse of $K^{(s)}$ that needs not be explicitly computed [4], $R^{(s)} = \text{Ker}(K^{(s)})$ is the null space of $K^{(s)}$, and $\alpha^{(s)}$ is a vector of six or fewer constants. The introduction of the additional unknowns $\alpha^{(s)}$ is compensated by the additional equations resulting from (4)

$$(6) \quad R^{(s)T} (F^{(s)} - B^{(s)T} \lambda) = 0 \quad s = 1, \dots, N_f$$

Substituting Eq. (5) into the second of Eqs. (2) and using Eq. (6) leads after some algebraic manipulations to the following FETI interface problem

$$(7) \quad \begin{bmatrix} F_I & -G_I \\ -G_I^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix}$$

where

$$(8) \quad \begin{aligned} F_I &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} B^{(s)T}; \\ G_I &= [B^{(1)} R^{(1)} \dots B^{(N_f)} B^{(N_f)}]; \\ \alpha^T &= [\alpha^{(1)T} \dots \alpha^{(N_f)T}]; \\ d &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} F^{(s)}; \\ e^T &= [F^{(1)T} B^{(1)} \dots F^{(N_s)T} B^{(N_f)}] \end{aligned}$$

$$(9) \quad \begin{aligned} K^{(s)+} &= K^{(s)-1} && \text{if } \Omega^{(s)} \text{ is not a floating substructure} \\ K^{(s)+} &= \text{a generalized inverse of } K^{(s)} && \text{if } \Omega^{(s)} \text{ is a floating substructure} \end{aligned}$$

For structural mechanics and structural dynamics problems, F_I is symmetric because the substructure matrices $K^{(s)}$ are symmetric. The objective is to solve by a PCG algorithm the interface problem (7) instead of the original problem (1). The PCG algorithm is modified with a projection enforcing that the iterates λ_k satisfy (6). Defining the projector P using

$$(10) \quad P = I - G_I (G_I^T G_I)^{-1} G_I^T$$

the algorithm can be written as:

$$(11) \quad \boxed{\begin{aligned} 1. \text{ Initialize} \\ \lambda^0 &= G_I (G_I^T G_I)^{-1} e \\ r^0 &= d - F_I \lambda^0 \\ 2. \text{ Iterate } k = 1, 2, \dots \text{ until convergence} \\ w^{k-1} &= P^T r^{k-1} \\ z^{k-1} &= \overline{F_I^{-1}} \bar{w}^{k-1} \\ y^{k-1} &= P \bar{z}^{k-1} \\ \zeta^k &= y^{k-1T} w^{k-1} / y^{k-2T} w^{k-2} \quad (\zeta^1 = 0) \\ p^k &= y^{k-1} + \zeta^k p^{k-1} \quad (p^1 = y^0) \\ \nu^k &= y^{k-1T} w^{k-1} / p^{kT} F_I p^k \\ \lambda^k &= \lambda^{k-1} + \nu^k p^k \\ r^k &= r^{k-1} - \nu^k F_I p^k \end{aligned}}$$

4. Data organization on DSM computer architecture

To be able to efficiently organize data for the FETI solver, we need to examine how the operating system will distribute memory inside the physical memory units and how this distribution affects the cost of accessing that memory. Simple observations of the impact of the computer architecture will give us guidelines to organize the elements involved in the FETI solver. The single distributed shared memory model of DSM machines simplifies the writing of parallel codes. However programmers must be conscious that the cost of accessing memory pages is not uniform and depends on the actual location in hardware of a page being accessed. On the SGI Origin 2000 machine, the operating systems distributes pages of memory onto physical pages by a *first touch* rule. This means that if possible, a memory page is allocated in the local memory of the first CPU that touches the page.

Fortunately, the FETI method, because it is decomposition based, lends itself in a natural way to a distribution of data across CPUs that guarantees that most of the memory is always accessed by the same CPU. To achieve this, one simply applies a distributed memory programming style on a shared memory architecture. This means that all operations relative to a subdomain s are always executed by the same CPU. This way, such objects as the local stiffness matrix $K^{(s)}$ will be created, factored and used for resolution of linear systems always on the same CPU.

5. Parallel programming paradigm

One can easily see that the operations involved in the FETI method are mostly matrix and vector operations that can be performed subdomain-per-subdomain. Such quantities as the residual vector or search direction vectors can be thought of as global quantities made of the assembly of subvectors coming from each subdomain. Operations on such vectors such as sum or linear combinations can be performed on a subdomain per subdomain basis. On the other hand, coefficients such as ν_k and ζ_k are scalar values which are global to the problem. These coefficients ensue mostly from dot products. Dot products can be performed by having the dot product of the subparts of the vectors performed by each subdomain in parallel, and then all the contributions summed up globally.

Such remarks have led us to write the program with a single thread executing the main PCG loop. In that way, only one CPU allocates and computes the global variables such as ν_k and ζ_k . To perform parallel operations, this single thread creates logical *tasks* to be performed by each CPU, and these tasks are distributed to as many parallel threads as CPUs being used. Examples of tasks are the assembly or factorization of $K^{(s)}$, or the update of the subpart of a vector for subdomain s .

Because the number of threads is arbitrary and independent of the number of tasks to be performed at a given step — i.e. several tasks can be assigned to the same thread —, independence between the number of subdomains and the number of CPUs is trivially achieved. In the extreme case, all tasks are executed by a single thread — the main thread — and the program can run on a sequential machine.

6. Implementation of the projector P

The application of the projector P is often referred to as the *coarse problem* because it couples all subdomains together. The application of the projector to a vector z can be seen as a three step process:

- Compute $\gamma = G^t z$
- Solve $(G^t G)\alpha = \gamma$
- Compute $y = z - G\alpha$

The first and last operation can be obtained in parallel with a subdomain per subdomain operation, since the columns of G (the trace of the rigid body modes) are non zero only for one subdomain interface. The second operation however is a global operation that couples all subdomains together.

Past implementations of the projector at the University of Colorado have relied on an iterative solution of the second equation of Eqs (6). Though such an approach was justified by the fact that these implementations were targeted at distributed memory machines, an experimental study of the problem has revealed that on DSM machine, it is more economical to solve this system with a direct solver. This direct solver can only be parallelized for a low number of CPU before losing performance.

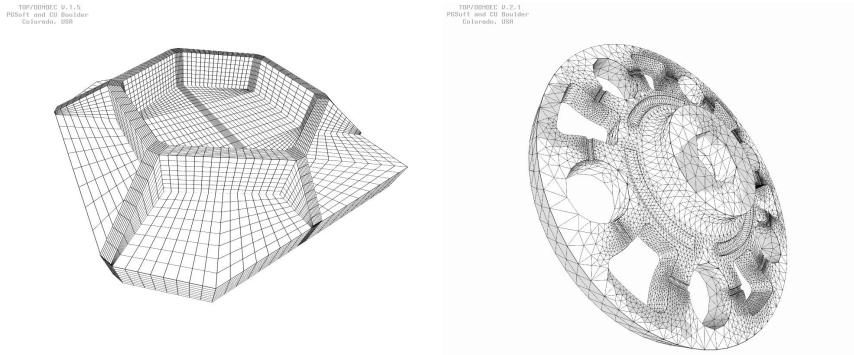


FIGURE 1. Finite element models of a lens (left) and a wheel carrier (right)

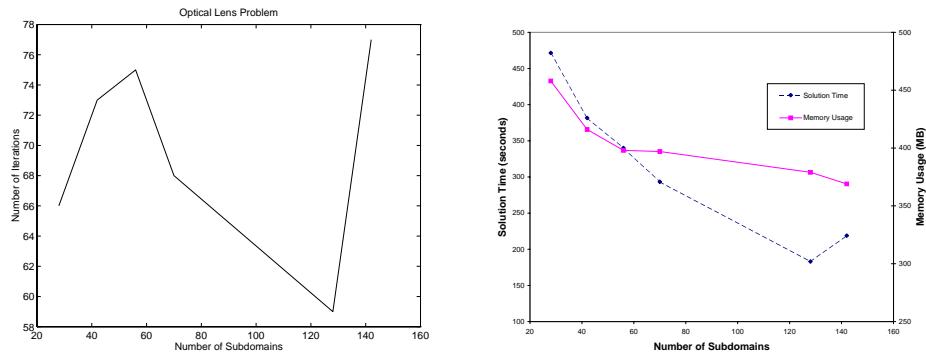


FIGURE 2. Number of iterations, solution time and memory requirements vs number of subdomains (Lens problem, 1 CPU)

Consequently, it is the least parallelizable part of the FETI solver and sets an upper limit to the performance that can be attained by adding CPUs.

7. Numerical experimentations

We have run our code on two significant example problems. For each problem we made runs with a varying number of subdomains and have recorded both timing of the solution and the memory required by the solver.

7.1. Lens problem. The first problem, shown in Fig. 1 on the left has 40329 nodes, 35328 brick elements and 120987 degrees of freedom. Solving this problem sequentially using a skyline solver and renumbered using RCM uses 2.2GB of memory and 10,000 seconds of CPU time. By contrast, on the same machine, running FETI sequentially with 128 subdomains requires 379MB of memory and 183.1 seconds of CPU. This results dramatically highlights that FETI is an excellent solution method, even on sequential machines. As can be seen on the left of Fig. 2, the number of iterations remains stable as the number of subdomains increases. The right part of the same figure shows the variation of timings and memory usage

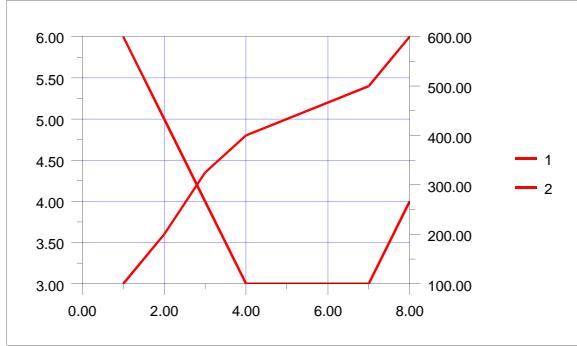


FIGURE 3. Solution time vs number of processors (lens problem, 128 subdomains)

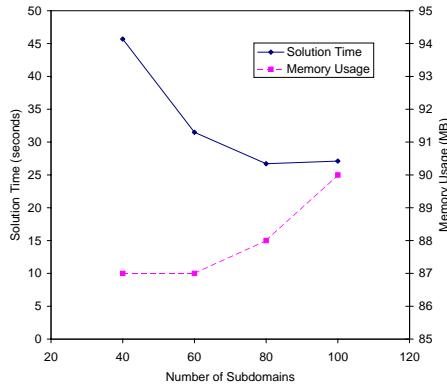


FIGURE 4. Solution time and memory requirements vs number of subdomains (Wheel carrier problem, 1CPU)

with the number of subdomains. It can be noticed that as the number of subdomains increases, the memory required to store all the local stiffness matrices $K^{(s)}$ decreases. This is mainly due to the reduction of the bandwidth of the local problems. Our experimentations show that the optimum decomposition for CPU time has 128 subdomains. Such a number would make it impractical for most users to use FETI with an implementation that requires one CPU per subdomain. After determining the optimum number of subdomains, we ran the same test case with an increasing number of processors. The resulting timings show good scalability (Fig. 3)

7.2. Wheel carrier problem. The second problem is a wheel carrier problem (see Fig. 1 on the right) with 67768 elements, 17541 nodes and 52623 degrees of freedom. The skyline solver requires 576 MB of memory and 800 seconds of CPU time. Fig. 4 shows a single CPU performance of FETI with 80 subdomains of 26.7s. On this problem, the memory requirement beyond 40 subdomains is stable around 88MB but tends to slightly increase as the number of subdomains increases. This is explained by the fact that as the number of subdomains increases, the size of the

interface increases and the memory required to store larger interface vectors offsets the reduction of memory required by the local stiffness matrices $K^{(s)}$.

8. Conclusions

We have implemented the FETI method on Distributed Shared Memory machines. We have achieved independence of the number of subdomains with respect to the number of CPUs. This independence has allowed us to explore the use of a large number of CPUs for various problems. We have seen from this experimentation that using a relatively large number of subdomains (around 100) can be very beneficial both in solution time and in memory usage. With such a high number of subdomains, the FETI method was shown to require CPU times and memory usage that are almost two orders of magnitude lower than those of a direct skyline solver. This strongly suggests that the FETI method is a viable alternative to direct solvers on medium size to very large scale problems.

References

1. C. Farhat, *A Lagrange multiplier based divide and conquer finite element algorithm*, J. Comput. Sys. Engrg. **2** (1991), 149–156.
2. C. Farhat, *A saddle-point principle domain decomposition method for the solution of solid mechanics problems*, Proc. Fifth SIAM Conference on Domain Decomposition Methods for Partial Differential Equations (D.E. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs, and R.G. Voigt, eds.), SIAM, 1991, pp. 271–292.
3. C. Farhat, J. Mandel, and F.X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Meths. Appl. Mech. Engrg. **115** (1994), 367–388.
4. C. Farhat and F.X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Meths. Engrg. **32** (1991), 1205–1227.
5. ———, *An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems*, SIAM J. Sc. Stat. Comput. **13** (1992), 379–396.
6. ———, *Implicit parallel processing in structural mechanics*, Computational Mechanics Advances **2** (1994), 1–124.

DEPARTMENT OF AEROSPACE ENGINEERING AND SCIENCES AND CENTER FOR AEROSPACE STRUCTURES UNIVERSITY OF COLORADO AT BOULDER BOULDER, CO 80309-0429, U.S.A.

Additive Schwarz Methods with Nonreflecting Boundary Conditions for the Parallel Computation of Helmholtz Problems

Lois C. McInnes, Romeo F. Susan-Resiga, David E. Keyes,
and Hafiz M. Atassi

1. Introduction

Recent advances in discretizations and preconditioners for solving the exterior Helmholtz problem are combined in a single code and their benefits evaluated on a parameterized model. Motivated by large-scale simulations, we consider iterative parallel domain decomposition algorithms of additive Schwarz type. The preconditioning action in such algorithms can be built out of nonoverlapping or overlapping subdomain solutions with homogeneous Sommerfeld-type transmission conditions on the artificially introduced subdomain interfaces. Generalizing the usual Dirichlet Schwarz interface conditions, such Sommerfeld-type conditions avoid the possibility of resonant modes and thereby assure the uniqueness of the solution in each subdomain.

The physical parameters of wavenumber and scatterer diameter and the numerical parameters of outer boundary diameter, mesh spacing, subdomain diameter, subdomain aspect ratio and orientation, subdomain overlap, subdomain solution quality (in the preconditioner), and Krylov subspace dimension interact in various ways in determining the overall convergence rate. Many of these interactions are not yet understood theoretically, thus creating interest in experimental investigation. Using the linear system solvers from the Portable Extensible Toolkit for Scientific Computation (PETSc), we begin to investigate the large parameter space and recommend certain effective algorithmic “tunings” that we believe will be valid for (at least) two-dimensional problems on distributed-memory parallel machines.

The external Helmholtz problem is the basic model of farfield propagation of waves in the frequency domain. This problem is challenging due to large discretized system sizes that arise because the computational grid must be sufficiently refined

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65F10, 65N30, 65Y05.

Supported by U.S. Department of Energy, under Contract W-31-109-Eng-38.

Supported in part by National Science Foundation grant ECS-9527169.

Supported in part by National Science Foundation grant ECS-9527169 and by NASA Contracts NAS1-19480 and NAS1-97046.

Supported in part by National Science Foundation grant ECS-9527169.

throughout the entire problem domain to resolve monochromatic waves, which requires approximately 10–20 gridpoints per wavelength for commonly used second-order discretizations. Moreover, the conventional farfield closure, the Sommerfeld radiation condition, is accurate only at large distances from centrally located scatterers and must be replaced at the artificial outer boundary by a nonreflecting boundary condition to obtain a computational domain of practical size. To this end we employ a Dirichlet-to-Neumann (DtN) map on the outer boundary, which provides an exact boundary condition at finite distances from the scatterer. The DtN map is nonlocal but does not introduce higher order derivatives.

Although the discretized Helmholtz linear system matrix is sparse, for a large number of equations direct methods are inadequate. Moreover, the Helmholtz operator tends to be indefinite for practical values of the wavenumber and the mesh parameter, leading to ill-conditioning. As a result conventional iterative methods do not converge for all values of the wavenumber, or may converge very slowly. For example, resonances can occur when conventional Schwarz-based preconditioners are assembled from Dirichlet subdomain problems. In view of these difficulties, this work focuses on developing a family of parallel Krylov-Schwarz algorithms for Helmholtz problems based on subdomain problems with approximate local transmission boundary conditions.

It is difficult to do justice to previous work on a century-old problem that has been revisited with vigor by specialists in diverse application areas in recent years. However, we select a few references that have been of inspirational value to our own work. Keller & Givoli [13] and Harari & Hughes [10] employed the global Dirichlet-to-Neumann map (a pseudo-differential operator) as a non-reflecting BC for the truncated domain (circle or sphere) and also experimented with truncating the complexity implied by the full DtN map. Després in his doctoral dissertation [7] pioneered domain decomposition for Helmholtz problems with first-order transmission conditions on nonoverlapping interfaces between subdomains, proving convergence to a unique solution. Ghanemi [8] combined nonlocal transmission conditions with Després-style iteration. She also obtained a better rate of convergence through under-relaxation of the nonoverlapping interface conditions. Douglas & Meade [12] advocated second-order local transmission conditions for both subdomain interfaces and the outer nonreflecting boundary condition and employed underrelaxed iterations. Our colleagues in domain-decomposed Helmholtz research, Cai, Casarin, Elliott & Widlund [3] introduced Schwarz-style overlapping, used first-order transmission conditions on the overlapped interfaces, calling attention to the *wavelap* parameter, which measures the number of wavelengths in the overlap region. They have also noted the importance of a (relatively fine) coarse grid component in the Schwarz preconditioner to overcome the elliptic ill conditioning that arises asymptotically for small mesh spacing.

2. Mathematical Formulation

The scalar Helmholtz equation,

$$(1) \quad -\nabla^2 u - k^2 u = 0,$$

is derived by assuming a time-harmonic variation in the solution of the second-order, constant-coefficient wave equation. A discussion of the hierarchy of models that reduce in their purest form to (1) is given in [1]. The parameter k is the reduced wavenumber, i.e. $2\pi/\lambda$, where λ is the wavelength. In the general case

anisotropy and spatially varying coefficients may be present, although in this work we restrict attention to a homogeneous isotropic problem.

We explicitly consider only the external Helmholtz problem, in which the perturbation field u is driven by a boundary condition inhomogeneity on a nearfield boundary Γ , subdivided into Dirichlet, Γ_D , and Neumann, Γ_N , segments, one of which may be trivial. Scatterer boundary conditions of sufficient generality are

$$(2) \quad u = g_D \text{ on } \Gamma_D \quad \text{and} \quad \partial u / \partial \nu = g_N \text{ on } \Gamma_N.$$

The Sommerfeld radiation condition,

$$\lim_{r \rightarrow \infty} r^{(d-1)/2} \left(\frac{\partial u}{\partial r} + ik u \right) = 0,$$

may be regarded as an expression of causality for the wave equation, in that there can be no incoming waves at infinity. Thus this condition acts as a filter that selects only the outgoing waves. (The Sommerfeld sign convention depends upon the sign convention of the exponent in the time-harmonic factor of the wave equation.)

The Dirichlet-to-Neumann Map. In finite computations the Sommerfeld boundary condition must be applied at finite distance. For B a circle (or a sphere in 3D), an integro-differential operator may be derived that maps the values of u on the artificial exterior boundary, B , to the normal derivative of u on B [9]. In two-dimensional problems this leads to an infinite series involving Hankel functions, which may be differentiated in the radial direction and evaluated at $r = R$ to yield on B :

$$(3) \quad \frac{\partial u}{\partial \nu}(R, \theta) \equiv (\mathcal{M}u)(R, \theta) = \frac{k}{\pi} \sum_{n=0}^{\infty} \frac{H_n^{(2)'}(kR)}{H_n^{(2)}(kR)} \int_0^{2\pi} \cos n(\theta - \theta') u(R, \theta') d\theta'.$$

This expression defines the Dirichlet-to-Neumann map, \mathcal{M} , where the infinite sum may be truncated to a finite approximation of $N \geq kR$ terms [10].

Finite Element Discretization. We use a Galerkin finite element formulation with isoparametric four-noded quadrilateral elements to discretize the problem specified by (1), (2), and (3) and thereby form a linear system,

$$(4) \quad Au = b.$$

Details about this system, as well as elementary properties of its pseudospectrum, are given in [14]. The Sommerfeld boundary condition has the effect of pushing the portion of the real spectrum that is close to (or at) zero in the Dirichlet case away from the origin, in the imaginary direction.

3. Schwarz-based Solution Algorithms

Brought into prominence in the current era of cache-based architectures, additive Schwarz methods have become the “workhorses” of parallel preconditioners for elliptically dominated partial differential equations in the last decade, and have recently been applied to Helmholtz problems in [3]. Although a variety of Schwarz-based techniques have been considered for Helmholtz problems by ourselves and others, this chapter focuses on accelerated overlapping iterative methods for the solution of the discrete equations (4).

Closely related to the overlapping Krylov-Schwarz method presented herein is a nonoverlapping stationary iterative scheme of Resiga and Atassi [16] for solving (1) independently in subdomains. This approach, which follows the work of Després

[7] and Ghanemi [8], uses under-relaxed impedance-type boundary conditions on subdomain interfaces and the DtN map on the exterior boundary, and features concurrent update of all subdomains for parallelism. Details of the scheme are presented in [14], where it is compared with the method featured herein.

We investigate an additive Schwarz preconditioner based on overlapping subdomains, which is accelerated by a Krylov method, such as a complex-valued version of GMRES [15]. An overlapping decomposition is defined by splitting the computational domain Ω into nonoverlapping subdomains Ω_i , with boundaries $\partial\Omega_i$, and extending each except where cut off by Γ and B to subdomains Ω'_i and interfaces $\partial\Omega'_i$. A (Boolean) restriction operator R'_i extracts those elements of the global vector that are local to extended subdomain i , and a prolongation operator R_i^T (without the ') prolongs elements local to the nonextended subdomain i back to the global vector, by extension with zeros. Let $A_i^{-1}u_i$ denote the action of solving in each extended subdomain,

$$\begin{aligned}\mathcal{L}_i u_i &= f_i \text{ in } \Omega'_i, \\ \mathcal{B}_i u_i &= \begin{cases} 0 & \text{on } \partial\Omega'_i - \partial\Omega \\ g_i & \text{on } \partial\Omega_i \cap \partial\Omega \end{cases}.\end{aligned}$$

Then a Schwarz projection M_i is defined by $M_i = R_i^T A_i^{-1} R'_i A$, and a Schwarz-preconditioned operator is then defined through $M = \sum_i M_i$. The system (4) is replaced with $Mu = \sum_i R_i^T A_i^{-1} R'_i b$. We iterate on this system with a Krylov method until convergence. This particular combination of extended restriction and unextended prolongation operators is designated “Restricted Additive Schwarz” (RAS) [4] and has been found by us and by others to be superior to standard Additive Schwarz, in which the prolongation is carried out to the extended subdomains. We have described the left-preconditioned form of RAS above. In practice, when comparing preconditioners as in this paper, we employ right-preconditioning, so that the residual norms available as a by-product in GMRES are not scaled by the preconditioner.

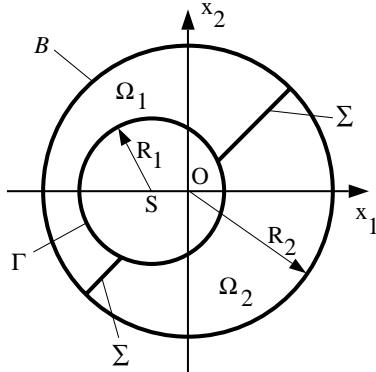
Each interior point of the original domain remains an interior point of at least one subdomain, and a standard PDE discretization is applied there. The extended interior subdomain interfaces are handled with Sommerfeld-type boundary conditions *in the preconditioner only*. Except for the use of homogeneous Sommerfeld-type boundary conditions, this method falls under the indefinite Schwarz theory of Cai & Widlund [5, 6].

4. A Model Helmholtz Problem

We use a model problem with a known exact solution to study the truncation error of this algorithm, along with the algebraic convergence rate. This model problem was employed by Givoli and Keller [9] for their demonstration of the advantages of the DtN map. The geometry and notation are defined in Figure 1. The eccentricity of the bounding circles spoils the application on B of simple boundary conditions based on normal incidence.

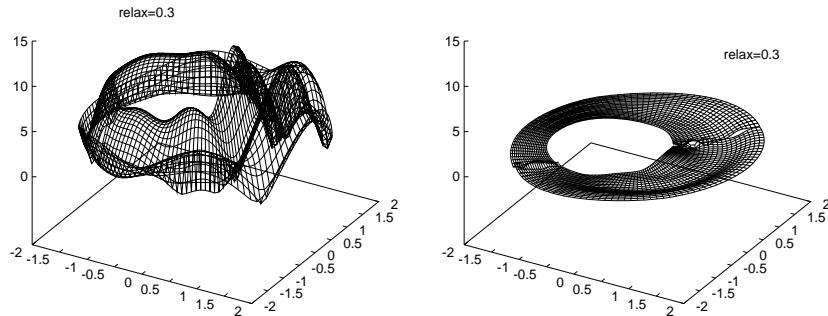
The explicit analytical solution u^* permits tabulation of the pointwise relative error in the numerical solution u_i^m at point i after iteration m as follows:

$$(5) \quad e_i^m = \frac{|u_i^m - u^*(x_i)|}{|u^*(x_i)|}.$$



- Inner boundary, Γ :
circle centered at $S = (-\frac{1}{2}, 0)$
with radius $R_1 = 1$
- Outer boundary, B :
circle centered at $O = (0, 0)$
with radius $R_2 = 2$
- Neumann BC on Γ from monopole at S
- Dirichlet-to-Neumann map on B

FIGURE 1. Eccentric annulus model problem domain.

FIGURE 2. Relative error distribution of converged solution with Sommerfeld (left) and the DtN map (right) exterior boundary conditions (nonoverlapping stationary iterative method, $k = 4$).

On the basis of numerical experiments that are discussed in [14], we have determined the truncation error ‘‘floor’’ beneath which we need not obtain algebraic convergence of (4). We have also compared a variety of schemes for attaining that level of error. Figure 2 shows a typical plot of the relative error (5) using a nonoverlapping stationary scheme for a two-subdomain case with $k = 4$. This picture emphasizes the well known advantage of a perfectly nonreflecting DtN map over a Sommerfeld boundary condition.

5. Numerical Results

For our numerical simulations of the Helmholtz problem, we employ the Portable, Extensible Toolkit for Scientific Computing (PETSc) [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. One feature that distinguishes it from other freely available libraries of iterative methods is the capability of solving systems defined over the complex numbers. Also, PETSc’s preconditioners make it routine to vary the number of subdomains in each physical dimension into which

TABLE 1. Iteration counts and parallel execution times (in seconds) for different aspect ratio subdomains and different overlaps, 65×256 grid, 32 processors.

Subdomain Shape		Overlap $h/2$		Overlap $3h/2$		Overlap $5h/2$		Overlap $9h/2$	
Procs	Subgrid	Its	Time	Its	Time	Its	Time	Its	Time
$k = 6.5$									
1×32	8:1	131	1.35s	125	1.38s	125	1.46	124	1.77s
2×16	2:1	137	1.54s	128	1.42s	128	1.89s	126	1.45s
4×8	1:2	174	1.96s	145	1.54s	138	1.77s	129	1.79s
8×4	1:8	211	2.98s	169	3.54s	153	2.44s	134	2.57s
$k = 13.0$									
1×32	8:1	159	1.92s	152	1.97s	150	1.99s	146	2.13s
2×16	2:1	182	1.88s	157	1.76s	150	1.83s	147	1.92s
4×8	1:2	195	1.92s	164	1.73s	157	2.08s	153	2.27s
8×4	1:8	224	2.84s	190	2.55s	176	2.73s	158	3.01s

the problem will be partitioned, the amount of overlap between these subdomains, and the quality of the solution process employed on each block in the preconditioner.

5.1. Comparison of Subdomain Shape and Overlap. Like convection problems, Helmholtz problems possess “preferred” directions in that there are dominant directions of wave propagation. Unlike convection problems, these directions are not manifest in the interior equations, which are locally rotationally invariant, but enter through the boundary conditions. It is therefore of interest to study the effect of subdomain size and shape on decomposed preconditioners for Helmholtz problems. We seek to answer two questions initially for a range of wavenumbers k : how does the orientation of the cuts interact with the orientation of the waves, and how much does overlap help to “pave over” the cuts?

Our implementation permits any number of radial and circumferential cuts, provided that all subdomains consist of a rectangular subset of the radial and circumferential indices. For the purpose of playing with aspect ratio in several increments over a large ratio, we select power-of-two size discretizations. Thus, we take 64 mesh cells in the radial direction and 256 in the circumferential. To satisfy the conservative $\lambda/h \geq 20$ in all directions throughout the domain, where $\lambda = 2\pi/k$, we consider $k = 6.5$. We compare this with $k = 13.0$, in which the waves are resolved with only 10 points per wavelength in the worst-resolved part of the domain (near $(x, y) = (2, 0)$). The action of A_i^{-1} is approximated on each subdomain by application of ILU(1), with overlap as tabulated across the column sets, accelerated by restarted GMRES. Wall-clock execution times are measured on 32 nodes of an IBM SP with 120MHz quad-issue Power2 nodes with a 10^{-4} relative residual tolerance.

We readily observe in Table 1 that cuts along constant angle (which are aligned with the dominant radial direction of wave propagation) are preferable over cuts along constant radius. Convergence is very much faster with few “bad” cuts than it is with many “bad” cuts. Overlap is effective in reducing the number of iterations by about 50% in the case of many “bad” cuts, but exhibits a relatively rapid law of diminishing returns in all orientations. Since the cost per iteration rises approximately linearly in the overlap and the convergence rate benefit saturates, the

TABLE 2. Scalability for fixed global problem size, 129×512 grid, $k = 13$

Processors	Iterations	Time (Sec)	Speedup	% Efficiency
1	221	163.01	—	—
2	222	81.06	2.0	100
4	224	37.36	4.4	100
8	228	19.49	8.4	100
16	229	10.85	15.0	93
32	230	6.37	25.6	80

experimentally observed optimal overlaps (corresponding to the italicized entries in the table) are all relatively modest.

5.2. Parallel Scalability. There are several measures of parallel scalability. Two of the most important are fixed-size scalability, in which more processors are employed in solving a problem of constant size, and fixed-memory-per-node (or “Gustafson”) scalability, in which a problem’s dimension and processor granularity are scaled in proportion. For the same algorithm, we employ a finer mesh of 128 cells radially and 512 angularly and we increase the wavenumber to $k = 13$ for this fixed-size problem. As shown in Table 2, we achieve overall efficiencies of 80% or better as the number of processors increases from 1 to 32. Convergence rate suffers remarkably mildly as preconditioner granularity increases. In this fixed-size problem, the algebraic dimension of the dense matrix block corresponding to the DtN map is fixed and is equally apportioned among the processors in a sectorial decomposition.

In the Gustafson scaling, in which the overall algebraic dimension of the problem grows in proportion to the number of processors, the communication involving all exterior boundary processors that is needed to enforce the DtN map implicitly has a deleterious effect on the scaled performance. We are presently addressing this problem by means of a sparsified approximation to the DtN map. The resulting operator is still much more accurate than a purely local Sommerfeld condition, but less crippling than the full global operator. For present purposes, we present the Gustafson scaling for a problem in which the DtN map is not included in the system matrix in (4), but split off to an outer iteration.

Results are shown in Table 3, over a range of three bi-dimensional doublings. Over one million grid points are employed in the finest case. As the problem is refined, we preserve the distribution of the spectrum by scaling with hk constant (fixed number of mesh points per wavelength). It could be argued that to keep the dominating phase truncation error term uniform, we should scale with $hk^{3/2}$ constant [11]. This would make k grow less rapidly in Table 3.

We obtain a reasonable *per iteration* efficiency, but we suffer a convergence rate degradation that is Poisson-like: iteration count grows as \sqrt{P} , where $P =$ number of subdomains. To remedy this problem, a (relatively fine) coarse grid [3] should be used in the Schwarz preconditioner.

6. Conclusions and Future Work

We have presented a parallel algorithm of Additive Schwarz type with Sommerfeld interface conditions for the wave Helmholtz problem. The benefits of DtN

TABLE 3. Scalability for fixed local problem size using an explicit implementation of the DtN map

Number of Processors	Global Dimension	k	Iters	Time per Iteration	
				Seconds	% Increase
4	129×512	13	250	.077	—
16	257×1024	26	479	.084	9
64	513×2048	52	906	.102	32

vs. Sommerfeld conditions on the exterior boundary are illustrated by comparison with analytical solution on model problem. Parallel scalability has been evaluated and is customarily good for an additive Schwarz method for a fixed-size problem. Relatively small overlaps are sufficient. The implicit DtN map, though highly accurate, intrinsically requires communication among all exterior boundary processors and hence is nonscalable, so sparsifications are under investigation. Without a global coarse grid, algorithmic scalability deteriorates in a Poisson-like manner as the mesh and processor granularity are refined.

This work is encouraging, but not definitive for parallel Helmholtz solvers. We are interested in better preconditioners to address the underlying elliptic convergence problems, and we are interested in higher-order discretizations to increase the computational work per grid point and reduce memory requirements for the same level of accuracy. Future work will include extensions to three-dimensional problems on less smooth domains, the addition of coarse grid to preconditioner, and the embedding of a Helmholtz solver in a multiphysics (Euler/Helmholtz) application in aeroacoustics.

Acknowledgements

The authors have benefitted from many discussions with X.-C. Cai, M. Casarin, F. Elliott, and O. B. Widlund.

References

1. H. M. Atassi, *Unsteady aerodynamics of vortical flows: Early and recent developments*, Aerodynamics and Aeroacoustics (K. Y. Fung, ed.), World Scientific, 1994, pp. 119–169.
2. S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *The Portable, Extensible Toolkit for Scientific Computing, version 2.0.21*, Tech. report, Argonne National Laboratory, 1997, <http://www.mcs.anl.gov/petsc>.
3. X.-C. Cai, M. Casarin, F. Elliott, and O. B. Widlund, *Schwarz methods for the Helmholtz problem with Sommerfeld boundary conditions*, Proceedings of the 10th International Conference on Domain Decomposition Methods (J. Mandel, C. Farhat, and X.-C. Cai, eds.), AMS, 1998.
4. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for nonsymmetric linear systems*, Tech. Report CU-CS-843-97, Computer Science Dept., Univ. of Colorado at Boulder, August 1997, http://www.cs.colorado.edu/cai/public_html/papers/ras_v0.ps.
5. X.-C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Comput. **13** (1992), 243–258.
6. _____, *Multiplicative Schwarz algorithms for nonsymmetric and indefinite elliptic problems*, SIAM J. Numer. Anal. **30** (1993), 936–952.
7. B. Després, *Méthodes de décomposition de domaines pour les problèmes de propagation d'ondes en régime harmonique*, Tech. report, University of Paris IX, 1991.
8. S. Ghanemi, *A domain decomposition method for Helmholtz scattering problems*, Proceedings of the 9th International Conference on Domain Decomposition Methods (P. E. Bjørstad, M. Espedal, and D. E. Keyes, eds.), Wiley, 1998.

9. D. Givoli, *Non-reflecting boundary conditions*, J. Comp. Phys. **94** (1991), 1–29.
10. I. Harari and T. J. R. Hughes, *Analysis of continuous formulations underlying the computation of time-harmonic acoustics in exterior domains*, Comp. Meths. Appl. Mech. Eng. **97** (1992), 103–124.
11. F. Ihlenberg and I. Babuška, *Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation*, Int. J. Numer. Meths. Eng. **38** (1995), 3745–3774.
12. J. Douglas Jr. and D. B. Meade, *Second-order transmission conditions for the Helmholtz equation*, Proceedings of the 9th International Conference on Domain Decomposition Methods (P. E. Bjørstad, M. Espedal, and D. E. Keyes, eds.), Wiley, 1998.
13. J. B. Keller and D. Givoli, *Exact non-reflecting boundary conditions*, J. Comp. Phys. **82** (1989), 172–192.
14. L. C. McInnes, R. F. Susan-Resiga, D. E. Keyes, and H. M. Atassi, *A comparison of Schwarz methods for the parallel computation of exterior Helmholtz problems*, In preparation, 1998.
15. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **7** (1986), 856–869.
16. R. F. Susan-Resiga and H. M. Atassi, *A domain decomposition method for the exterior Helmholtz problem*, J. Comp. Phys., To appear, 1998.

MATHEMATICS AND COMPUTER SCIENCE DIVISION, ARGONNE NATIONAL LABORATORY, ARGONNE, IL 60639-4844

E-mail address: curfman@mcs.anl.gov

AEROSPACE & MECHANICAL ENGINEERING DEPARTMENT, UNIVERSITY OF NOTRE DAME, NOTRE DAME, IN 46556

E-mail address: rsusanre@light.ame.nd.edu

COMPUTER SCIENCE DEPARTMENT, OLD DOMINION UNIVERSITY, NORFOLK, VA 23529-0162 & ICASE, NASA Langley Res. Ctr., HAMPTON, VA 23681-2199

E-mail address: keyes@cs.odu.edu

AEROSPACE & MECHANICAL ENGINEERING DEPARTMENT, UNIVERSITY OF NOTRE DAME, NOTRE DAME, IN 46556

E-mail address: atassi@carmen.ame.nd.edu

On the Reuse of Ritz Vectors for the Solution to Nonlinear Elasticity Problems by Domain Decomposition Methods

Franck Risler and Christian Rey

1. Introduction

This paper deals with a Rayleigh-Ritz Preconditioner (RRP) that accelerates convergence for the iterative solution to a series of symmetric positive definite linear systems associated with nonlinear substructured elasticity problems. RRP depends upon CG's superconvergent properties and consists of a suitable reuse of Ritz vectors. Moreover, the Rayleigh-Ritz paradigm can be wisely associated with another acceleration technique, the Generalized Krylov Correction, so as to form the SPARKS (Spectral Approach for the Reuse of Krylov Subspaces) algorithm. Numerical assessment of both RRP and SPARKS is provided on a large-scale poorly-conditioned engineering practice.

2. Solution to Nonlinear Elasticity Problems

We consider computation of the equilibrium of bodies made up of compressible hyperelastic material and that undergo large deformation. A Lagrangian formulation is chosen and all variables are defined in the reference configuration. Moreover, Ω in \mathbb{R}^3 and Γ exhibit the domain occupied by the body and its boundary respectively. The equilibrium equations may then be written in a weak form as follows

$$(1) \quad \left\{ \begin{array}{l} \text{Find } u \in \{H + u_0\} \text{ such that} \\ \int_{\Omega} \frac{\partial \Phi}{\partial F}(u) : \nabla v d\Omega = \int_{\Omega} f.v d\Omega + \int_{\Gamma_g} g.v d\Gamma \quad \forall v \in H \\ H = \{v \in H^1(\Omega)^3, v = 0 \text{ on } \Gamma_u = \Gamma - \Gamma_g\} \end{array} \right.$$

where H denotes the space of kinematically-admissible displacement fields, $(:)$ stands for the double contractor operator between two tensors A and B ($A : B = \text{Tr}(A^T B)$), x are the coordinates of any particle of the domain measured in the reference configuration (Ω) in a fixed orthonormal basis of \mathbb{R}^3 , u_0 is the imposed

1991 *Mathematics Subject Classification*. Primary 65B99; Secondary 65Y05.

Key words and phrases. Domain Decomposition, Nonlinear Elasticity, Superlinear Convergence, Krylov Subspaces, Ritz Values.

Supercomputing facilities were provided by the IDRIS Institute (Institut du Développement et de Ressources en Informatique Scientifique). The two authors are grateful to the IDRIS Institute for this support.

displacement field on part Γ_u of the domain boundary, $v(x)$ is any admissible displacement field in the reference configuration, $u(x)$ is the unknown displacement field, $F(x) = Id + \nabla u(x)$ is the deformation gradient, $g(x)$ is the surface tractions on part Γ_g of the domain boundary, complementary to Γ_u in Γ , $f(x)$ is the density of body forces (we assume that external loadings f and g do not depend on the displacement field u - dead loading assumption), and Φ is the specific internal elastic energy.

The problem given by Eq.(1) is discretized through a finite element method [13] and leads to the solution to a nonlinear problem of the form $\mathcal{F}(u) = 0$. Such a discrete problem is solved by means of Newton-type methods that amount to the resolution to a succession of symmetric positive definite linear problems, the right hand sides and the matrices of which are to be reactualized.

The reader may refer to [1] for a complete presentation of nonlinear elasticity problems. Moreover, further explanations on Newton-type algorithms can be found in [5] or in [6].

3. Iterative Solution to a Series of Linear Problems

3.1. The Substructuring Paradigm. By condensing each linear problem on the subdomains interface, non overlapping Domain Decomposition (DD) methods (primal [7] or dual [4] approach) enable to solve iteratively with a Conjugate Gradient (CG) algorithm the following succession of linear problems,

$$(2) \quad (P^k) : A^k x^k = b^k \quad , \quad k = 1, \dots, m$$

where A^k denotes the matrix of Schur complement either in primal or dual form depending on the approach chosen, and b^k is the associated condensated right hand sides. Note that the A^k matrix herein considered is symmetric, positive, definite [9]. From now on, we will be focusing on the dual domain-decomposition paradigm. The proposed Ritz preconditioner may nevertheless suit to the primal approach, though some characteristic properties of the dual interface operator magnify the positive effects upon the convergence of this preconditioner.

3.2. Definition and Fundamental properties.

3.2.1. *CG characterizing Properties.* The Conjugate Gradient algorithm applied to the solution to the linear problem (P^k) arising from Eq.(2), depends on the construction of a set of w_i^k descent directions that are orthogonal for the dot product associated with the A^k matrix. The Krylov subspace thus generated may be written as

$$(3) \quad K_{r_k}(A^k) = \{ w_0^k, w_1^k, \dots, w_{r_k-1}^k \} \quad ; \quad K_{r_k}(A^k) \subset \mathbb{R}^n$$

where the subscript r_k denotes the dimension of this latter subspace and n exhibits the number of unknowns of the substructured problem to be solved.

A characteristic property of CG is given by

$$(4) \quad \|x^k - x_{r_k}^k\|_{A^k} = \min_{y \in x_0^k + K_{r_k}(A^k)} \|x^k - y^k\|_{A^k}$$

where x_0^k is a given initial field and with $\|v\|_{A^k} = (A^k v, v)$.

Consequently, by introducing the A^k -orthogonal projector $P_{K_{r_k}}^{A^k}$ onto the $K_{r_k}(A^k)$ Krylov subspace, the r_k -rank approximation of the solution can be written

$$(5) \quad x_{r_k}^k = x_0^k + P_{K_{r_k}}^{A^k}(b^k - A^k x_0^k) \quad \text{with} \quad P_{K_{r_k}}^{A^k}(x) = \sum_{i=0}^{r_k-1} \frac{(x, w_i^k)}{(A^k w_i^k, w_i^k)} w_i^k$$

3.2.2. Ritz Vectors and Values. The Ritz vectors $y_j^{(r_k)} \in K_{r_k}(A^k)$ and $\theta_j^{(r_k)} \in \mathbb{R}$ values are defined such that [8]

$$(6) \quad A^k y_j^{(r_k)} - \theta_j^{(r_k)} y_j^{(r_k)} \perp K_{r_k}(A^k)$$

The convergence of the Ritz values towards a set of r_k eigenvalues of the A^k matrix exhibits the dominating phenomenon, on which the CG's rate of convergence and the so-called superlinear convergence behavior [8, 12] depends.

4. The Rayleigh-Ritz Preconditioner

4.1. A Krylov Based Spectral Approach. The purpose of this new preconditioner is to utilize spectral information related to the dominating eigenvalues arising from Krylov subspaces so as to accelerate the resolution of a succession of linear systems of the form given by Eq.(2). The relevance of this approach has been analysed in ([11], criterion 2.3) and its validity domain has been defined.

More precisely, we intend herein to very significantly accelerate the convergence of a set of p dominating Ritz values to trigger a superconvergent behavior of CG. The key to the Ritz approach lies in the so-called *effective* condition number that quantitatively weights the rate of the CG's convergence in the course of the resolution process. The *effective* condition number is defined at the j iteration of the CG as the ratio of the largest uncaptured eigenvalue of the A^k matrix to its smallest eigenvalue. Note that a given λ eigenvalue of the A^k matrix is considered captured whenever a Ritz value provides a sufficiently accurate approximation of λ so that the corresponding eigenvector no longer participates in the solution process [12].

Therefore, with the Ritz approach, we seek to drastically reduce the *effective* condition number within the first CG's iterations. Besides, the spectrum of the dual interface operator is distinguished by few dominating eigenvalues which are not clustered and are well separated from the smaller ones [4]. Consequently, the new algorithm is expected to be even more efficient than some of the intrinsic spectral properties of the linear problems we deal with, magnify its positive effects upon convergence of the dominating Ritz values.

Let us define a $Q \in \mathbb{R}^{n \times p}$ matrix, the columns of which store an approximation of p eigenvectors of the current A^k operator. Inasmuch as we are aiming to reduce the *effective* condition number, we will prescribe at the i iteration of the Conjugate Gradient algorithm an optional orthogonality constraint that is presented as

$$(7) \quad Q^T g_i = 0 \quad \forall i$$

In terms of Krylov subspaces, that yields

$$(8) \quad K_{r_k}(A^k) \subset \text{Ker } Q^T = (\text{Im } Q)^\perp$$

Note that this orthogonality constraint is similar to the one associated with a new framework that has been recently introduced to speed up convergence of

dual substructuring methods [3]. But, while in [3] considerations upon domain decomposition method found the algorithm, the Ritz approach depends on a spectral analysis of the condensed interface matrix. Consequently, apart from this sole formulation similarity, these two methods are based on completely different concepts.

Furthermore, we emphasize the fact that the constraint given by Eq.(7) is optional and, for obvious reasons, does not modify the admissible space to which the solution belongs. Besides, providing that the constraint is enforced at each CG's iteration, it must consequently be verified by the solution to the linear problem. Since the residual vector associated with the final solution is theoretically equal to zero, the orthogonality condition prescribed by Eq.(7) is thus satisfied.

Let's now focus on the construction of the Q matrix in the framework of the resolving to a series of linear problems. We advocate that the approximation of eigenvectors arises from the Ritz vectors associated with the p dominating Ritz values originating from the first system (P^1). Note that the efficiency of the conditioning problem depicted in Eq.(7) is submitted to two main assumptions ([11], Hypothesis 3.1) in (a) the convergence of Ritz vectors, and (b) the perturbation of eigendirections among the family $\{A^k\}_{k=1}^{k=m}$ of matrices.

Moreover, if the number of linear problems to be solved is high and the columns of the Q matrix do not provide a sufficiently accurate approximation of eigenvectors related to dominating eigenvalues of a given A^q matrix ($1 < q \leq m$), the Q matrix has to be reactualized. Hence, it requires suspending the prescription of the constraint given in Eq.(7) while solving (P^q) and computing the Ritz vector associated with the p dominating Ritz values in order to update the Q matrix. The Ritz conditioning problem is then restored until another reactualization procedure is required.

4.2. Construction of the Rayleigh-Ritz Preconditioner. For the sake of clarity, and since no confusion is possible, the k superscript is herein omitted and the A^k matrix is simply noted A .

In order to prescribe the optional constraint given by Eq.(7), we shall superpose, at each iteration i of the Conjugate Gradient algorithm, the field x_i and an additional field of Lagrange multipliers ξ_i such that

$$(9) \quad \begin{aligned} x_i &\longrightarrow \tilde{x}_i = x_i + \xi_i = x_i + Q\alpha_i \\ Q^T \tilde{g}_i &= 0 \quad \text{with} \quad \tilde{g}_i = A\tilde{x}_i - b \end{aligned}$$

Substituting the second equation of Eq.(9) into the first one yields

$$(10) \quad Q^T A^k Q \alpha_i + Q^T A^k x_i - Q^T b^k = 0$$

Consequently, α is given by

$$(11) \quad \alpha_i = -(Q^T A^k Q)^{-1} Q^T A^k x_i + (Q^T A^k Q)^{-1} Q^T b^k$$

Then, substituting Eq. (11) into Eq.(9) yields

$$(12) \quad \begin{aligned} \tilde{x}_i &= (I - Q(Q^T A^k Q)^{-1} Q^T A^k)x_i + Q(Q^T A^k Q)^{-1} Q^T b^k \\ \hat{x}_i &= (I - Q(Q^T A^k Q)^{-1} Q^T A^k)x_i + x_0 \\ \text{with } x_0 &= Q(Q^T A^k Q)^{-1} Q^T b^k \end{aligned}$$

Let the projector \bar{P} be defined by

$$(13) \quad \bar{P}: x_i \longrightarrow \tilde{x}_i - x_0 \quad \text{with} \quad \bar{P} = (I - Q(Q^T A^k Q)^{-1} Q^T A^k)$$

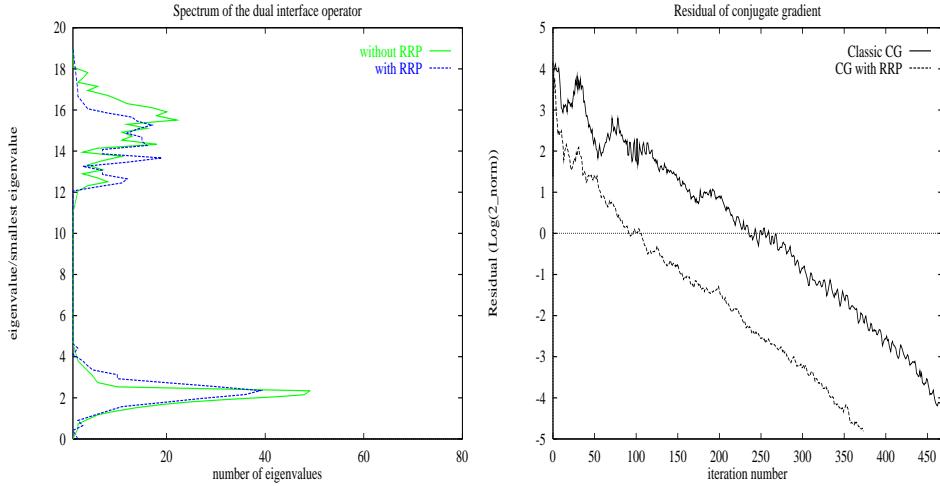


FIGURE 1. Spectral distribution and residual history with or without the RRP (R100) preconditioner

Since the considered projector \bar{P} is not symmetric, the prescription of the Rayleigh-Ritz Preconditioner (RRP) at each CG iteration has then to be performed in two projection steps, by \bar{P}^T and \bar{P} respectively.

5. Application

Numerical efficiency is assessed on a large-scale poorly-conditioned non linear problem: a three-dimensional steel-elastomer laminated structure that distinguishes with great heterogeneousness and high nonlinearity. The considered structure has a parallelepiped geometry and is discretized by hexaedral finite elements (Q1 elements). Besides, an axial compression loading with an imposed displacement is applied. Material behavior is modelled by the Ciarlet-Geymonat specific internal energy Φ [2] and the associated equivalent Young modulus and Poisson coefficient are $(E, \nu) = (1.3 \text{ MPa}; 0.49)$ and $(E, \nu) = (2 \times 10^5 \text{ MPa}; 0.3)$ for the elastomer and the steel respectively.

On account of the quadratic convergence of the Newton methods, the stopping criterion of Newton iterations (nonlinear iterations) is set to 10^{-6} while the accuracy requirement for solving each linear problem is 10^{-3} . In all cases, the linear problems are also preconditioned by the classical *Lumped* preconditioner [4] and the spectral results have been estimated from the Ritz values for the Krylov space generated by the Conjugate Gradient algorithm, when a number of iterations n_r (close to the dimension n of the problem) is performed. Finally, N_s and N denote from now on the number of processors and the number of unknowns of the nonlinear problem considered respectively.

5.1. Numerical Performance. In Table 1, is reported the number of iteration achieved by the Conjugate Gradient algorithm within the Newton iterations and the Figure 1 exhibits the spectral distribution and the residual history of the (P^2) linear problem. In all cases, *Classic CG* means that RRP is not applied and *CG with RRP* (R_p) indicates that a RRP whose size is p is prescribed. On the

TABLE 1. Numerical performances with various preconditioners

N_s	N	solver	Newton Iterations		
			N_1	N_2	N_3
15	85680	Classic CG	294	307	367
15	85680	CG with RRP (R50)	294	254	312
15	85680	CG with RRP (R100)	294	212	272
15	85680	CG GKC	294	120	71
15	85680	CG with SPARKS (R50)	294	73	51

spectral distribution, we observe that, not only the condition number κ of the RRP preconditioned matrix is reduced, but also the dominating values are fewer and more spread out. It thus paves the way for a fast-convergence of Ritz values towards the dominating eigenvalues – and hence a drastic reduction of the *effective* condition number – during the first CG iterations, what is supported by the chart of the residual history. On the other hand, this latter curve shows that, in a second phase of the resolution process, when Ritz values have converged towards the eigenvalues associated with the eigenvectors, an approximation of which is provided by the columns of the Q matrix, the rate of convergence is decelerated and becomes close to the one of the Classical CG.

Computational results in terms of CPU time are not addressed in this paper since they highly depend on implementation issues and would require further explanations. Nonetheless, numerical assessments show that the provided acceleration of convergence is not offset against the computational overheads involved by the RRP prescription.

5.2. The SPARKS algorithm. Whereas the Rayleigh-Ritz preconditioner is based on a condensation of information related to the upper part of the spectrum, the Generalized Krylov Correction[10] reuses in a broader and a non-selective way information originating from previously generated Krylov subspaces. We associate those two latter algorithm within an hybrid (RRP-GKC) preconditioner, which we will be calling from now on SPARKS (Spectral Approach for the Reuse of Krylov Subspaces).

Numerical experiments show that the RRP has a dominating contribution during the first iterations, when the Conjugate Gradient (CG) explores spectral subspaces related to the dominating eigenvalues. Afterwards, the rate of convergence is mainly ruled by the GKC preconditioner which enables to speed up the capture of phenomena associated with lower frequencies. A numerical assessment is provided in Table 1 and further validations have shown that the very significant acceleration of convergence provided by RRP within the SPARKS algorithm goes far beyond the computational overcost generated. Moreover, SPARKS distinguishes with an increasing efficiency when the size n of the problem grows.

6. Conclusion

We have presented in this paper a Raleigh-Ritz preconditioner, that is characterized by the reuse of spectral information arising from previous resolution processes. Principles and construction of this preconditioner have been addressed

and numerical performance of the Ritz approach has been demonstrated on a large-scale poorly-conditioned engineering problem. Moreover, a new hybrid Krylov-type preconditioner, known as SPARKS, and deriving from both the Rayleigh-Ritz and the Generalized Krylov Correction has been introduced and has proved outstanding numerical performances.

References

1. P.G. Ciarlet, *Mathematical elasticity*, North-Holland, Amsterdam, 1988.
2. P.G. Ciarlet and G. Geymonat, *Sur les lois de comportement en elasticité non linéaire compressible*, C.R. Acad. Sci. Paris **T. 295, Série II** (1982), 423–426.
3. C. Farhat, P-S. Chen, F. Risler, and F-X. Roux, *A simple and unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers*, International Journal of Numerical Methods in Engineering (1997), in press.
4. C. Farhat and F-X. Roux, *Implicit parallel processing in structural mechanics*, vol. 2, Computational Mechanics Advances, no. 1, North-Holland, june 1994.
5. H.B. Keller, *The bordering algorithm and path following near singular points of higher nullity*, SIAM J. Sci. Stat. Comput. **4** (1983), 573–582.
6. P. Le Tallec, *Numerical analysis of equilibrium problems in finite elasticity*, Tech. Report CEREMADE 9021, Cahier de la décision, Université Paris-Dauphine, 1990.
7. ———, *Implicit parallel processing in structural mechanics*, vol. 1, Computational Mechanics Advances, no. 1, North-Holland, june 1994.
8. B. N. Parlett, *The symmetric eigenvalue problem*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
9. C. Rey, *Développement d’algorithmes parallèles de résolution en calcul non linéaire de structures hétérogènes: Application au cas d’une butée acier-elastomère*, Ph.D. thesis, Ecole Normale Supérieure de Cachan, 1994.
10. C. Rey and F. Léné, *Reuse of Krylov spaces in the solution of large-scale nonlinear elasticity problems*, Proceeding of the Ninth International Conference on Domain Decomposition Methods, Wiley and Sons, in press.
11. C. Rey and F. Risler, *The Rayleigh-Ritz preconditioner for the iterative solution of large-scale nonlinear problems*, Numerical Algorithm (1998), in press.
12. A. Van der Luis and H. A. Van der Vorst, *The rate of convergence of conjugate gradients*, Numerische Mathematik **48** (1986), 543–560.
13. O. Zienkiewicz, *The finite element method*, McGraw-Hill, New-York, Toronto, London, 1977.

LABORATOIRE DE MODÉLISATION ET MÉCANIQUE DES STRUCTURES - U.R.A 1776 DU C.N.R.S,
U.P.M.C - TOUR 66 - 5IÈME ETAGE - 4, PLACE JUSSIEU 75252 PARIS CEDEX 05

E-mail address: risler@rip.ens-cachan.fr, rey@ccr.jussieu.fr

Dual Schur Complement Method for Semi-Definite Problems

Daniel J. Rixen

1. Introduction

Semi-definite problems are encountered in a wide variety of engineering problems. Most domain decomposition methods efficient for parallel computing are based on iterative schemes and rarely address the problem of checking the problem's singularity and computing the null space. In this paper we present a simple and efficient method for checking the singularity of an operator and for computing a null space when solving an elliptic structural problem with a dual Schur complement approach.

The engineering community has long been reluctant to use iterative solvers mainly because of their lack of robustness. With the advent of parallel computers, domain decomposition methods received a lot of attention which resulted in some efficient, scalable and robust solvers [3, 6]. The Finite Element Tearing and Interconnecting method (FETI) has emerged as one of the most useful techniques and is making its way in structural and thermal commercial softwares [1, 4].

So far, the issue of semi-definite problems in FETI has not been fully addressed although a broad range of engineering problems are singular. For instance, the static and vibration analysis of satellites, aircrafts or multi-body structures is governed by [5]

$$(1) \quad Ax = b$$

where A is a symmetric semi-definite positive stiffness matrix, x are the structural displacements and b is the vector of external forces. The zero energy modes u_i , $i = 1, \dots, m$, define a null space such that

$$(2) \quad Au_i = 0 \quad i = 1, \dots, m$$

and a solution exists for problem (1) only if

$$(3) \quad u_i^T b = 0 \quad i = 1, \dots, m$$

When using direct solvers to solve a singular problem such as (1), the null space is obtained as a by-product of the factorization when detecting zero pivots [5]. Unfortunately, when iterative solvers are applied, the algorithms do not provide any

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65F10, 15A03, 73C02.
The author was supported by the Belgian National Science Foundation.

information on the singularity of the problem and the null space is never computed in the iteration process.

In this paper, a general Conjugate Gradient procedure for testing the singularity of an operator and for extracting the null space is presented. Iterative methods for computing a null space exist (e.g. singular value decomposition or inverse iteration with spectral shifting [5]), but they entail a tremendous computational cost when applied to large systems. Here we show how algorithms like Conjugate Gradient for solving linear systems can be adapted to check for singularity and to compute a null space, thereby adding only a small computational overhead and involving only minor alteration to the solution procedure. We will discuss its application to the FETI solver.

2. Finite element tearing and inter-connecting

2.1. The dual Schur complement formulation. The solution of a problem of the form (1) where A is a symmetric positive matrix arising from the discretization of some second- or fourth-order elliptic structural mechanics problem on a domain Ω , can be obtained by partitioning Ω into N_s substructures $\Omega^{(s)}$, and gluing these with discrete Lagrange multipliers λ [3]:

$$(4) \quad A^{(s)}x^{(s)} + B^{(s)T}\lambda = b^{(s)} \quad s = 1, \dots, N_s$$

$$(5) \quad \sum_{s=1}^{N_s} B^{(s)}x^{(s)} = 0$$

where the superscript (s) denotes a quantity pertaining to $\Omega^{(s)}$, $B^{(s)}$ is a signed Boolean matrix such that $B^{(s)}x^{(s)}$ is the restriction of $x^{(s)}$ to the subdomain interface boundary and λ are Lagrange multipliers associated to the interface compatibility constraints (5). From Eqs. (4), $x^{(s)}$ can be computed as

$$(6) \quad x^{(s)} = A^{(s)+} \left(b^{(s)} - B^{(s)T}\lambda \right) + R^{(s)}\alpha^{(s)}$$

where $A^{(s)+}$ denotes the inverse of $A^{(s)}$ if $\Omega^{(s)}$ is not singular, or a generalized inverse of $A^{(s)}$ otherwise. In the latter case, $R^{(s)} = \text{Ker}(A^{(s)})$ stores a basis of the null space of $A^{(s)}$ and is obtained during the factorization of $A^{(s)}$, and $\alpha^{(s)}$ stores the amplitudes of $R^{(s)}$. If $A^{(s)}$ is singular, (4) requires that

$$(7) \quad R^{(s)T} \left(b^{(s)} - B^{(s)T}\lambda \right) = 0$$

From Eqs. (6) and (5), and recalling condition (7), the interface problem can be written as [3]

$$(8) \quad \begin{bmatrix} F_I & -G_I \\ -G_I^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix}$$

where

$$(9) \quad \begin{aligned} F_I &= \sum_{s=1}^{N_s} B^{(s)}A^{(s)+}B^{(s)T}; & d &= \sum_{s=1}^{N_s} B^{(s)}A^{(s)+}b^{(s)} \\ G_I &= \begin{bmatrix} B^{(1)}R^{(1)} & \dots & B^{(N_s)}R^{(N_s)} \end{bmatrix}; & \alpha &= \begin{bmatrix} \alpha^{(1)T} & \dots & \alpha^{(N_s)T} \end{bmatrix}^T \\ e &= \begin{bmatrix} b^{(1)T}R^{(1)} & \dots & b^{(N_s)T}R^{(N_s)} \end{bmatrix}^T \end{aligned}$$

Splitting the Lagrange multipliers as

$$(10) \quad \lambda = \lambda_0 + P\bar{\lambda}$$

$$(11) \quad \text{where} \quad \lambda_0 = G_I(G_I^T G_I)^{-1} e$$

$$(12) \quad P = I - G_I(G_I^T G_I)^{-1} G_I^T$$

the interface problem (8) is transformed into the semi-definite system

$$(13) \quad (P^T F_I P) \bar{\lambda} = P^T (d - F_I \lambda_0)$$

$$(14) \quad \alpha = (G_I^T G_I)^{-1} G_I^T (d - F_I \lambda)$$

Hence, a solution of the original indefinite system of interface equations (8) can be obtained by applying a Preconditioned Conjugate Gradient (PCG) algorithm to the symmetric semi-definite interface problem (13). Such a procedure can also be viewed as a Preconditioned Conjugate Projected Gradient (PCPG) algorithm [3].

At every PCPG iteration, the projection steps require the solution of a coarse grid problem associated with the subdomain floating modes of the form

$$(15) \quad (G_I^T G_I) \alpha = G_I^T w$$

These coarse grid problems are solved by second level Conjugate Gradient iterations with a projection and re-orthogonalization technique [2, 3] in order to re-use the Krylov spaces computed at previous iterations.

2.2. FETI applied to semi-definite problems. When problem (1) is positive semi-definite, the null space directions u_i verifying (2) also satisfies the sub-domain-wise null space condition

$$(16) \quad A^{(s)} u_i^{(s)} = 0$$

$$(17) \quad \sum_{s=1}^{N_s} B^{(s)} u_i^{(s)} = 0$$

stating that the null space of the global problem is also a null space for the subdomains and satisfies the interface compatibility. Hence, at the subdomain level, the global null space vectors are linear combinations of the local floating modes, i.e

$$(18) \quad u_i^{(s)} = R^{(s)} \theta_i^{(s)}$$

and computing the null space for the global problem is equivalent to finding the amplitudes of the local null space directions such that

$$(19) \quad \sum_{s=1}^{N_s} B^{(s)} R^{(s)} \theta_i^{(s)} = G_I \theta_i = 0$$

where $\theta_i = [\theta_i^{(1)^T} \dots \theta_i^{(N_s)^T}]^T$. Therefore, when m null space vectors u_i exist, G_I is no longer full rank and the coarse grid operator $(G_I^T G_I)$ has a null space of dimension m such that

$$(20) \quad (G_I^T G_I) \theta_i = 0 \quad i = 1, \dots, m$$

In this case, a solution exists for the dual interface problem (8) only if e is in the range of G_I , i.e.

$$(21) \quad \theta_i^T e = 0 \quad i = 1, \dots, m$$

Expanding this condition further by using definition (9) yields

$$\sum_{s=1}^{N_s} \theta_i^{(s)T} R^{(s)T} b^{(s)} = \sum_{s=1}^{N_s} u_i^{(s)T} b^{(s)} = u_i^T b = 0 \quad i = 1, \dots, m$$

Thus, a solution exists for the dual interface problem if b is in the range of A in the initial problem (1) and in that case a starting value λ_0 can be computed by (11). A solution to the coarse problem (15) can be found by building a generalized inverse $(G_I^T G_I)^+$ if a direct solver is used. For distributed memory machines, an iterative algorithm is usually preferred: a non preconditioned Conjugate Gradient scheme can still be applied since the successive directions of descent remain in the range of $(G_I^T G_I)$.

However, it is important that the singularity of the coarse grid be detected in the FETI method and that the null space be computed in order to check for condition (21), otherwise the FETI iterations could proceed without converging.

3. Conjugate Gradient iterations for semi-definite problems

The method for checking the singularity of a symmetric semi-definite positive operator and for computing the associated null space consists in applying a Conjugate Gradient iteration scheme to

$$(22) \quad Ax = Ay$$

where $Ay \neq 0$.

If the initial guess for x is set to zero, the computed x will be in the range of A since the directions of descent of the Conjugate Gradient iteration are combinations of the residual of (22). Therefore, if the solution x at convergence is equal to y independently on the choice of y , we can state that A is non-singular. Otherwise, $u = x - y$ yields a null space vector. In the latter case, the iteration is restarted with a new y direction orthogonal to the null space already extracted: $x - y$ is now searched for in a deflated space.

Using a projection and re-orthogonalization technique for solving problems with multiple right-hand sides [2], the algorithm can be summarized as in Table 1. In this algorithm, the directions of descent stored in X are normalized such that $X^T W = I$, W storing the results of AX . Note that the null space of A is usually computed with a very high accuracy. Hence the stopping criterion for the Conjugate Gradient should be $\|r^k\| < \epsilon \|b\|$ with ϵ very small.

Clearly, two major issues remain to be cleared in algorithm 1, namely how to set the vectors y and what criterion to apply for the condition $x - y \neq 0$. Choosing y correctly is of crucial importance for the success of the algorithm: on one hand it should not be in the null space so that $Ay \neq 0$, and on the other hand, it should contain enough null space components so that $x - y = 0$ occurs only if all the null space vectors have been extracted. It is clear that the choice of y as well as the criterion for $x - y \neq 0$ should be based on an estimate of the condition number of A which can be gathered from the Conjugate Gradient coefficients. Nevertheless, in the next section we propose a simple and efficient technique for choosing y and checking for $x - y \neq 0$ when algorithm 1 is applied to the coarse grid problem 15 in FETI for structural problems.

TABLE 1. Iterations for extracting the null space of A

```

Setup  $y$ 
Set  $b = Ay$ 
 $x^0 = 0, r^0 = b$ 
Projection if  $X$  exists
 $x^0 = X(X^T b)$ 
 $r^0 = b - W(X^T b)$ 
Iterate for  $k = 0, \dots$ 
 $p^k = r^k - X(W^T r^k)$ 
 $n^k = p^{k^T} A p^k, \nu^k = p^{k^T} r^k / n^k$ 
 $x^{k+1} = x^k + \nu^k p^k$ 
 $r^{k+1} = r^k - \nu^k A p^k$ 
 $X = [X, p^k / \sqrt{n^k}], W = [W, A p^k / \sqrt{n^k}]$ 
End
If  $x - y \neq 0$ ,
 $u_i = x - y$ 
choose a new  $y$ 
orthogonalize  $y$  with respect to  $u_i, \forall i$ 
restart at  $b = Ay$ 

```

4. Preliminary coarse grid iterations in FETI

The computational cost of the procedure described in 1 can be significant since the number of descent directions for computing the entire null space with a good accuracy can be large. For instance, if applied to the iterative solution of the non decomposed problem (1), the singularity check would cost more than the actual computation of the solution.

As explained in section 2.2, the singularity issue for FETI appears in the coarse grid problem which dimension is very small compared to the dimension of u . Moreover, for the coarse grid problem, the complete Krylov space is needed anyway during the FETI iterations. Hence, solving a set of preliminary coarse grid problems of the form $(G_I^T G_I)x = (G_I^T G_I)y$ for finding the null space of $(G_I^T G_I)$ entails only a small overhead cost.

Note that for the primal Schur complement approach [6], algorithm 1 can be applied to the assembled Schur complement. This would however induce an unacceptable computational cost since it would amount to solving the entire interface problem several times. If however the balancing method version of the primal Schur method is used [7, 6], the singularity check can be performed at a low cost on the coarse grid operator associated to the Neumann preconditioner in a way similar to what is presented here for FETI.

Null vector criterion. To define a criterion for $x - y \neq 0$ in algorithm 1, we decompose y into

$$(23) \quad y = \bar{y} + \sum_{i=1}^m \theta_i \beta_i$$

where \bar{y} is the component of y in the range of $(G_I^T G_I)$. Since by construction the solution x is in the range of $(G_I^T G_I)$, $x - y \simeq \sum \theta_i \beta_i$. We then assume that the

successive starting vectors y are chosen to ensure a good representation of the null space, i.e. so that

$$(24) \quad \left\| \sum_{i=1}^m \theta_i \beta_i \right\| > \|y\|/N \quad \text{if } m \neq 0$$

when the dimension N of the coarse grid is not trivially small. Hence we state that $x - y \neq 0$ if

$$(25) \quad \|x - y\| > \|y\|/N$$

Choosing the starting vectors. Let us remind that G_I is the restriction of the local null spaces to the interface boundary and the columns of $R^{(s)}$ are usually orthonormalized so that they represent the rigid translational modes and the rotation modes around the nodal geometric centers (or around the center of gravity if the orthogonality is enforced with respect to the mass matrix). Since $G_I y$ represents the interface displacement jumps for local rigid body displacements of amplitude y , choosing

$$(26) \quad y_1 = [\text{diag}(G_I^T G_I)]^{-1} [1 \ 1 \ 1 \ 1 \ \dots]^T$$

all local translational and rotational modes around the local centers are included, and therefore the resulting displacement field is not compatible on the interface: $G_I y_1 \neq 0$. The initial vector y_1 in (26) is scaled by the diagonal of $(G_I^T G_I)$ in order to account for the fact that subdomains may have very different sizes thus different boundary displacements for the rotational modes.

Since y_1 is non-zero for all local rigid body modes, condition (24) is satisfied in practice. Based on similar mechanical considerations, the next initial vectors y are then chosen as follows

$$(27) \quad \begin{aligned} y_2 &= [\text{diag}(G_I^T G_I)]^{-1} [1 \ 0 \ 1 \ 0 \ \dots]^T \\ y_3 &= [\text{diag}(G_I^T G_I)]^{-1} [1 \ 0 \ 0 \ 1 \ \dots]^T \quad \dots \end{aligned}$$

The technique proposed in this paper for the FETI method has been implemented in the finite element analysis code SAMCEF and several free-free structures have been analyzed using this technique. In the next section we describe some of the test results.

5. Application examples

Free-free plane stress example. To illustrate the robustness of the proposed technique, let us first consider a square plane stress structural model decomposed into 8×8 square subdomains, each subdomain containing 10×10 finite elements (Fig. 1). No Dirichlet boundary conditions are applied to this two-dimensional problem so that every subdomain has 3 rigid body modes, the coarse grid problem is of dimension 192 and there are 3 global rigid body modes ($m = 3$).

Applying the iteration scheme 1 to the coarse grid operator $(G_I^T G_I)$ with a tolerance for the Conjugate Gradient iterations of $\epsilon = 10^{-14}$ yields the correct three global rigid body modes. The convergence of the scheme for extracting the successive θ_i is described in Fig. 1. The convergence curves show that the number of iterations decrease every time a new null space component is searched for due to the projection and re-orthogonalization steps. Let us remind the reader that the directions computed and stored in this pre-processing step are used later on when the

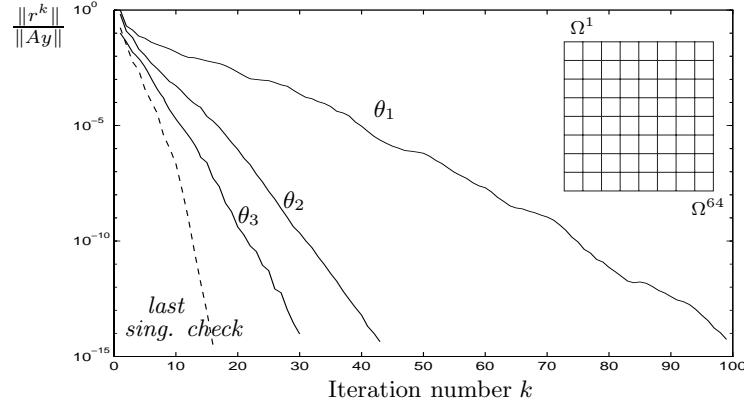


FIGURE 1. A free-free plane stress example

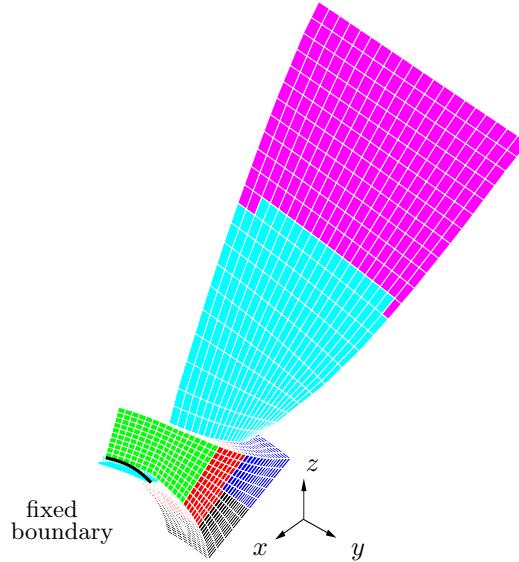


FIGURE 2. Decomposition of a blade model with 3 rigid body modes

linear system is actually solved by applying a projection and re-orthogonalization technique.

Analysis of a blade. We now present an example representing a realistic turbine blade modeled by brick elements. The model is decomposed into 6 subdomains and the displacements are fixed in the vertical direction on a curved edge at the base of the blade (Fig. 2). Note that because the fixed boundary is curved and belongs to the (z, y) plane, the vertical constraints not only restrain the vertical translation and the rotation about the x axis, but they also restrain the rigid rotation about the y axis. Hence only 3 global rigid body modes exist.

Since the fixed boundary is only slightly curved, the problem of detecting the local floating modes for the constrained substructure is badly conditioned. However

this does not affect the conditioning of the coarse grid problem. Applying algorithm 1, we found the exact 3 rigid body modes. Note that if the operator A of the entire structure would have been factorized, detecting the global rigid body modes would have been much more difficult.

6. Conclusion

In this paper we have addressed the problem of checking the singularity of a problem and computing the associated null space within the iterative solution procedure of a linear system. The method has been adapted to the FETI method in structural mechanics. A simple, low cost and robust technique has been proposed. It requires only preliminary iterations on the coarse grid problem associated to the subdomain null spaces and uses existing FETI technology. Hence our method entails only small modifications to the FETI algorithm and minor computational costs. The effectiveness of the procedure was demonstrated on some relevant examples. Equipped with this important singularity check, the FETI method can be used as an efficient solver in general static and free vibration analysis.

References

1. *Ansys powersolver*, USACM-Net Digest, July 27 1995.
2. C. Farhat, L. Crivelli, and F.X. Roux, *Extending substructures based iteratives solvers to multiple load and repeated analyses*, Comput. Methods Appl. Mech. Engrg. **117** (1994), 195–209.
3. C. Farhat and F. X. Roux, *Implicit parallel processing in structural mechanics*, Comput. Mech. Adv. **2** (1994), no. 1, 1–124, North-Holland.
4. M. Gérardin, D. Coulon, and J.-P. Delsemme, *Parallelization of the SAMCEF finite element software through domain decomposition and FETI algorithm*, Internat. J. Supercomp. Appl. **11** (1997), 286:298.
5. M. Gérardin and D. Rixen, *Mechanical vibrations, theory and application to structural dynamics*, 2d ed., Wiley & Sons, Chichester, 1997.
6. P. LeTallec, *Domain-decomposition methods in computational mechanics*, Comput. Mech. Adv. **1** (1994), 121–220, North-Holland.
7. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg. **9** (1993), 233–241.

DEPARTMENT OF APPLIED SCIENCE, LTAS, UNIVERSITY OF LIÈGE, 4000 LIÈGE, BELGIUM
Current address: Department of Aerospace Engineering Sciences, Center for Aerospace Structures, University of Colorado, Boulder CO 80309-0429

E-mail address: d.rixen@colorado.edu

Two-level Algebraic Multigrid for the Helmholtz Problem

Petr Vaněk, Jan Mandel, and Marian Brezina

1. Introduction

An algebraic multigrid method with two levels is applied to the solution of the Helmholtz equation in a first order least squares formulation, discretized by Q1 finite elements with reduced integration. Smoothed plane waves in a fixed set of directions are used as coarse level basis functions. The method is used to investigate numerically the sensitivity of the scattering problem to a change of the shape of the scatterer.

Multigrid methods for the solution of the Helmholtz equation of scattering are known in the literature. A common disadvantage of multigrid methods is that the coarsest level must be fine enough to capture the wave character of the problem, or the iterations diverge [6, 7, 10]. One way to overcome this limitation is to use coarse basis functions derived from plane waves [8]. However, manipulating such functions becomes expensive since the cost does not decrease with the number of variables on the coarse levels. It should be noted that functions derived from plane waves can also be used as basis functions for the discretization of the Helmholtz equation itself; such methods are known under the names of the Microlocal Discretization [5], Partition of Unity Finite Element Method [1], or the Finite Element Ray Method [11].

We propose a two-level method with coarse space basis functions defined as a plane waves within an aggregate of nodes, zero outside the aggregate, and then smoothed by a Chebyshev type iteration using the original fine level matrix. This results in a method with good computational complexity and scalability. This method falls under the abstract framework of black-box two-level iterative methods based on the concept of smoothed aggregations [16]. The objective of the smoothing of the coarse basis functions is to reduce their energy [9, 15, 16]. For a related theoretical analysis of such two-level methods with high order polynomial

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35J05.

Key words and phrases. Helmholtz Equation, Scattering, Algebraic Multigrid, First Order Least Squares, Perturbation of Domain.

This research was supported by the Office of Naval Research under grant N-00014-95-1-0663. Supercomputing facilities were provided by the Naval Research Laboratory and by the National Center for Supercomputing Applications. This paper is based on the unpublished report [17], <http://www-math.cudenver.edu/ccmreports/rep110.ps.gz>, where more computational results can be found.

smoothing of coarse basis functions, see [3], where a convergence result uniform both with respect to coarse and fine level meshsize was proved for second order elliptic problems discretized on unstructured meshes.

2. Problem Formulation and Discretization

In this section, we describe the first order least squares formulation and the discretization of the two-dimensional scattering problem on a bounded domain.

Denote by H^{div} the space of all vector functions with divergence in L^2 . Let $\mathcal{O} \subset \Omega$ be an obstacle in a domain $\Omega \subset \mathbb{R}^2$ that is sufficiently large with respect to the size of \mathcal{O} . To be more precise, Ω is assumed large enough to allow the scattered field to be almost radial near $\partial\Omega$.

We seek a complex pressure $p \in H^1(\Omega \setminus \mathcal{O})$ and its (complex) gradient $u \in H^{div}(\Omega \setminus \mathcal{O})$ minimizing the convex functional

(1)

$$F(p, \mathbf{u}) = w_1 \int_{\Omega \setminus \mathcal{O}} \|\nabla p - \mathbf{u}\|^2 + w_2 \int_{\Omega \setminus \mathcal{O}} |\operatorname{div}(\mathbf{u}) + k^2 p|^2 + w_3 \int_{\partial\Omega} |\mathbf{u} \cdot \mathbf{r} - ikp|^2,$$

where w_1, w_2, w_3 are positive constants and $\mathbf{r} \in \mathbb{R}^2$ is the normalized radiusvector of the point $\mathbf{x} \in \partial\Omega$, i.e. $\mathbf{r} = \mathbf{x}/\|\mathbf{x}\|$. The pressure p is subject to the Dirichlet boundary condition on the boundary of the obstacle,

$$p(\mathbf{x}) = -\exp(ik\mathbf{d} \cdot \mathbf{x}/\|\mathbf{d}\|), \quad \mathbf{x} \in \partial\mathcal{O},$$

where $\mathbf{d} \in \mathbb{R}^2$ is the direction of the incident wave. The first two integrals are a first order least square formulation corresponding to the Helmholtz equation

$$\Delta p + k^2 p = 0,$$

and the boundary integral enforces the radiation boundary condition

$$\frac{\partial p}{\partial \mathbf{r}} = ikp \quad \text{on } \partial\Omega.$$

Hence, all the integrals in (1) vanish for the minimizer of F and the solution of the minimization problem (1) is independent of the weights w_1, w_2, w_3 . We choose the uniform Q1 finite elements for the discretization of the continuous minimization problem, and minimize the functional over the finite element space. As the derivatives of the gradient of a Q1-function are not Q1-functions themselves, the discretization of the integrals in (1) creates an undesirable “artificial viscosity” resulting in a damping of the numerical solution. In fact, it is easy to see on a one-dimensional example, that the restriction of a plane wave on the mesh is not a solution of the discrete problem. To avoid this, the volume integrals have been discretized by the one point quadrature formula with the node in the middle of each element. This restores the property that the discrete system allows plane waves as solutions. Such *inexact integration* is frequently used to eliminate locking caused by similar integrals in plate and shell finite elements. Since $u = \nabla p$ satisfies $\nabla \times u = 0$, adding the term

$$(2) \quad \int_{\Omega \setminus \mathcal{O}} |\nabla \times u|^2$$

to the functional F in (1), as suggested in [4, 8], does not change the solution and makes the functional F coercive in certain cases.

In the discrete case, adding the integral (2) to F changes the discrete solution and may make it more accurate. We have not found an advantage to adding the integral (2) in our experiments. The results with reduced integration alone were satisfactory. Also, in the discrete case, the solution is no longer independent of the weights w_1, w_2, w_3 . Based on our computations, we found no significant advantage to other than unit weights, which are used in all computations reported here.

The discretization described above results in a Hermitian matrix with 3 complex degrees of freedom per node. Because we wanted to take advantage of an existing real code, our implementation treated the real and imaginary parts of the solution as real unknowns, resulting in a real symmetric problem with 6 real degrees of freedom per node. Further efficiency could be gained by an implementation in the complex arithmetics.

3. Algebraic Multigrid

In this section, we describe the algebraic method used for solving the discretized scattering problem. It is a variant of the method introduced in [3, 16]. We will present the method as a variant of the multiplicative Schwarz method [2, 12], and write it in terms of matrices.

3.1. The Multiplicative Schwarz Method. Let A be a symmetric, positive definite $n \times n$ matrix, and N_j , $j = 0, \dots, m$, be full rank matrices with n rows. Consider the following iterative method for the solution of the linear algebraic system $A\mathbf{x} = \mathbf{f}$:

$$\begin{aligned} (3) \quad & \mathbf{z} \leftarrow \mathbf{x}^i \\ (4) \quad & \mathbf{z} \leftarrow \mathbf{z} + N_i(N_i^T A N_i)^{-1} N_i^T (\mathbf{f} - A\mathbf{z}), \quad i = 1, \dots, m \\ (5) \quad & \mathbf{z} \leftarrow \mathbf{z} + N_0(N_0^T A N_0)^{-1} N_0^T (\mathbf{f} - A\mathbf{z}), \\ (6) \quad & \mathbf{z} \leftarrow \mathbf{z} + N_i(N_i^T A N_i)^{-1} N_i^T (\mathbf{f} - A\mathbf{z}), \quad i = m, \dots, 1 \\ (7) \quad & \mathbf{x}^{i+1} \leftarrow \mathbf{z} \end{aligned}$$

Since (3)-(7) is a consistent stationary iterative method for the system $A\mathbf{x} = \mathbf{f}$, it can be written as $\mathbf{x}^{i+1} = \mathbf{x}^i + N(\mathbf{f} - A\mathbf{x}^i)$. We use the operator N , that is, the output of (3)-(7) with $\mathbf{x}^i = 0$, as a preconditioner in the method of conjugate gradients. The operator N is symmetric and positive definite, since [2, 12]

$$(8) \quad NA = (I - \Pi_1) \cdots (I - \Pi_m)(I - \Pi_0)(I - \Pi_m) \cdots (I - \Pi_1),$$

where Π_i is the A -orthogonal projection onto the range of N_i .

3.2. Two-level Algebraic Multigrid by Smoothed Aggregation. It remains to specify the matrices N_i . We first construct the matrix N_0 . This matrix is denoted $N_0 = P$ and called the *prolongator*. The prolongator is constructed in two steps. In the first step, we construct a *tentative prolongator* capturing precisely a selected set of functions (in the same sense as, for example, P1 finite elements capture linear functions). In the second step, we suppress high-energy components in the range of the tentative prolongator by smoothing it using a proper *prolongator smoother*. The matrices N_i , $i \neq 0$, are injections that correspond to overlapping blocks of unknowns. The blocks are derived from the nonzero structure of the smoothed prolongator. As a prolongator smoother we use a properly chosen polynomial in the stiffness matrix A . The degree of the prolongator smoother

determines the smoothness of the coarse space as well as the amount of overlaps of the blocks in the overlapping Schwarz method.

To construct the prolongator P , we first decompose the set of all nodes, where an essential boundary condition is not imposed, into a disjoint covering

$$(9) \quad \{1, \dots, n\} = \bigcup_{i=1}^m \mathcal{A}_i, \quad \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ for } i \neq j.$$

We use a simple greedy algorithm that chooses aggregates of nodes that are connected via nonzero terms of A , whenever possible, cf. [13, 16].

Let us consider the set of functions $\{f^i\}_{i=1}^{n_k}$ we want to be captured by the coarse space functions. These functions should approximate the kernel of the constrained problem well (e.g., constants in the case of Poisson equation, 6 rigid body modes in the case of 3D elasticity). For solving the scattering problem here, our choice of the set of functions $\{f^i\}_{i=1}^{n_k}$ are the plane waves

$$f_{\mathbf{d}}(\mathbf{x}) = e^{-ik\mathbf{x} \cdot \mathbf{d}/\|\mathbf{d}\|},$$

where $\mathbf{x} \in \mathbb{R}^2$ is the position and $\mathbf{d} \in \mathbb{R}^2$ is the direction of the plane wave. We choose a finite subset of plane waves; the numerical experiments in this paper are performed with a coarse space built from plane waves in the $n_k = 8$ directions $\mathbf{d} = (1, 0), (-1, 0), (0, 1), (0, -1), (1, 1), (-1, -1), (-1, 1), (1, -1)$. Since plane waves are not contained in the fine level space exactly, the vectors $\{\hat{\mathbf{f}}^i\}_{i=1}^{n_k}$ are constructed as grid interpolations of the functions $f_{\mathbf{d}}$ and their gradients.

For each function f^i , let $\hat{\mathbf{f}}^i$ be its discrete representation in the finite element basis. Each node has 6 degrees of freedom, namely, the real and imaginary parts of p , $\partial_x p$, and $\partial_y p$. The decomposition (9) induces a decomposition of the degrees of freedom into disjoint sets

$$\{1, \dots, n_d\} = \bigcup_{i=1}^m \mathcal{D}_i, \quad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset \text{ for } i \neq j,$$

where \mathcal{D}_i is the set of all degrees of freedom associated with the nodes of \mathcal{A}_i . We then construct a *tentative prolongator* as the block matrix

$$(10) \quad \hat{P} = [\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_m]$$

where the j -th row of the block column $\hat{\mathbf{p}}_i$ equals to the j -th row of $\hat{\mathbf{f}}^i$ if $j \in \mathcal{D}_i$, and is zero otherwise. The prolongator P is then defined by

$$P = s(A)\hat{P},$$

where s is the polynomial of given degree d such that $s(0) = 1$ and $\max_{0 \leq \lambda \leq \hat{\rho}} |s(\lambda)|^2 \lambda$ is minimal, with $\hat{\rho}$ an easily computable upper bound on the spectral radius of A . It is easy to show [14] that p is a shifted and scaled Chebyshev polynomial, equal to

$$s(\lambda) = \left(1 - \frac{\lambda}{r_1}\right) \cdots \left(1 - \frac{\lambda}{r_d}\right), \quad r_k = \frac{\hat{\varrho}}{2} \left(1 - \cos \frac{2k\pi}{2n+1}\right).$$

This construction of P attempts to minimize the energy of the columns of P . Indeed, any column of the prolongator P is $s(A)\hat{\mathbf{p}}$, where $\hat{\mathbf{p}}$ is a column of the tentative prolongator \hat{P} , and its squared energy norm is

$$\|s(A)\hat{\mathbf{p}}\|_A^2 = (s(A)\hat{\mathbf{p}})^T A(s(A)\hat{\mathbf{p}}) = \hat{\mathbf{p}}^T s^2(A) A \hat{\mathbf{p}}.$$

TABLE 1. Model problem for performance experiments

Computational domain:	$[-2, 2] \times [-2, 2]$
Obstacle:	$[-0.3, 0.3] \times [-0.3, -0.3]$
Dir. of the inc. wave:	$\mathbf{d} = (1, 1)$
k	varying
Mesh:	regular 400×400 square mesh
Num. of dofs:	964,806

If the columns of \hat{P} have bounded euclidean norm (which they do here), the construction above gives an optimal bound on the energy of the columns of P , uniform in the choice of \hat{P} . See [3, 14] for more details, and [9] for a more direct approach to minimizing the energy of the columns of P .

The iteration (5) has now the interpretation of a *coarse grid correction*, in terms of multigrid methods. It remains to choose the matrices N_i in the *pre-smoothing* (4) and *post-smoothing* (6). Our choice is equivalent to a multiplicative overlapping Schwarz method. The overlapping blocks are derived from the nonzero structure of the prolongator P by choosing N_i to consist of those columns j of the n_d by n_d identity matrix, for which the j -the row of the block column \mathbf{p}_i from (10) is not zero. The iterations (4) and (6) are now simply block relaxation with overlapping blocks, with block i consisting of all variables that may become nonzero in the range of the prolongator block column \mathbf{p}_i .

3.3. Parallel Implementation. The smoothing iterations (4) and (6) are parallelized using a generalization of the well known red-black ordering for Gauss-Seidel iteration. First, observe that if $N_i^T A N_j = 0$, then the projections Π_i and Π_j from (8) commute, and so the iteration steps i and j in (4) or (6) are independent and can be performed concurrently.

The method proceeds as follows. In the setup phase, a coloring of the adjacency graph of the matrix $(N_i^T A N_j)_{i,j}$ is found by a simple greedy algorithm. (Of course, the graph is constructed by performing symbolic matrix multiplication only.) Then $N_i^T A N_j = 0$ for all indices i and j assigned the same color. The smoothing iteration (4) is then reordered so that all iteration steps with the same color are done concurrently. Then the pre-smoothing (4) becomes

$$\mathbf{z} \leftarrow \mathbf{z} + \left(\sum_{i \in \mathcal{C}_k} N_i (N_i^T A N_i)^{-1} N_i^T \right) (\mathbf{f} - A\mathbf{z}), \quad k = 1, \dots, n_c,$$

where n_c the number of colors and \mathcal{C}_k is the set of all indices of color k . The post-smoothing (6) is same except that the colors are processed in the reverse order.

4. Performance Results

All experiments reported in this section have been carried out on a SGI ORIGIN 2000 with 64 R10000 processors and 4GB of memory. As the stopping condition, we have used

$$\langle AP\mathbf{r}^i, A\mathbf{r}^i \rangle^{1/2} \leq \frac{10^{-5}}{\text{cond}} \langle AP\mathbf{r}^0, A\mathbf{r}^0 \rangle^{1/2},$$

where P is the preconditioner, cond is a condition number estimate computed from the conjugate gradient coefficients, and \mathbf{r}^i is the residual after the i -th iteration. In order to illustrate the effect of this stopping condition, we provide achieved relative

TABLE 2. Performance of the method with varying k . H denotes the side of the aggregates (squares). Mesh 400×400 elements. Number of degrees of freedom 964,806. Number of processors 16. $H/h = 30$.

k	els./wave length	$H/\text{wave length}$	cond.	achieved rel. residual	setup time [s] CPU/WALL	iter. time [s] CPU/WALL
251.327	5	6.0	6.565	2.447×10^{-6}	49/64	115/131
125.663	10	3.0	15.63	1.670×10^{-6}	48/63	117/118
83.7758	15	2.0	31.67	1.729×10^{-6}	52/71	169/171
62.8318	20	1.5	60.70	1.112×10^{-6}	53/73	243/246
50.2654	25	1.2	103.9	8.939×10^{-7}	48/64	330/333
41.8879	30	1.0	170.9	7.271×10^{-7}	50/65	446/451

TABLE 3. Performance of the method depending on the size H of the aggregates. Fixed $k = 125.6637$. Mesh 400×400 elements. Number of degrees of freedom 964,806. Number of processors 8.

H/h (els)	$H/\text{wave length}$	cond.	memory [MB]	setup [s] CPU/WALL	iter [s] CPU/WALL
10	1.0	36.63	1,245	232/248	332/335
15	1.5	26.80	1,088	91/108	210/213
20	2.0	20.98	1,077	74/87	169/171
25	2.5	17.18	1,055	75/88	155/155
30	3.0	15.63	1,106	71/86	155/157
35	3.5	13.88	1,140	77/92	144/145
40	4.0	12.48	1,204	93/108	155/154
45	4.5	11.06	1,237	90/106	136/138
50	5.0	10.32	1,298	104/119	140/142

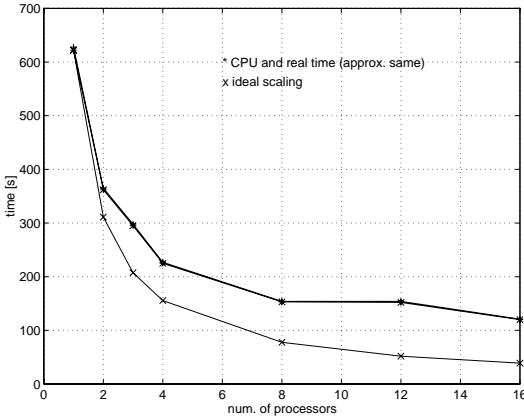
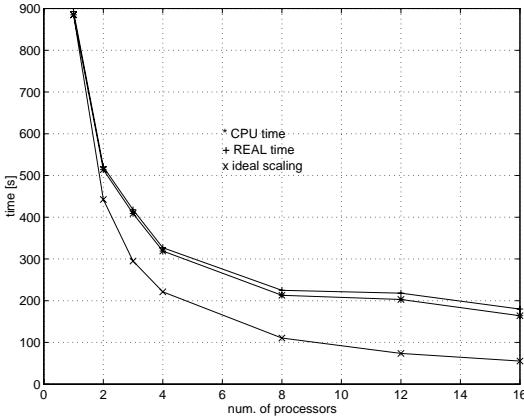
residuals (measured in Euclidean norm) in Table 2. In all experiments presented here we used a prolongator smoother of degree 4.

Three different types of experiments were done: testing the method in the case of varying k , varying the coarse space size, i.e. the size of the aggregates \mathcal{A}_i , and parallel scalability. In all experiments, we have used the scattering model problem described in Table 1.

Due to the regular geometry of our testing problem, we were able to use a system of square aggregates. The size of the side of those squares is denoted by H .

Our results are summarized in Tables 2 and 3, and graphs in Figures 1 and 2. In all the experiments, the actual iteration history was well characterized by the condition number of the preconditioned matrix (we observed no spectral clustering). In other words, the error was being reduced by the factor of about $(\sqrt{\text{cond}} - 1)/(\sqrt{\text{cond}} + 1)$ per iteration. The runtime condition number estimates are reported.

We observe that the condition number of the preconditioned problem depends on H/λ_k , the ratio of the subdomain size to the wavelength $\lambda_k = 2\pi/k$. The condition number decreases with this ratio increasing (due either to increasing the frequency or to enlarging the subdomain size). We offer the following heuristic explanation: In the case of Helmholtz equation, the low-energy functions are waves

FIGURE 1. Parallel scalability of the iteration phase. $k = 125.6637$, $H = 30h$.FIGURE 2. Parallel scalability of the method (setup+iter). $k = 125.6637$, $H = 30h$.

with wavelength close to the parameter λ_k . For the problem with λ_k small compared to the subdomain size, the direct subdomain solvers (playing the role of a smoother) are capable of approximating such functions well.

The performance tests have shown that using 8 processors, we gain parallel speedup of about 4 times, and the effect of adding more processors seems to be decreasing. We believe that this scalability issue reflects the early stage of parallelization of the code at the time of writing.

References

1. I. Babuška and J. M. Melenk, *The partition of unity finite element method*, Int. J. Numer. Meths. Engrg. **40** (1997), 727–758.
2. Petter E. Bjørstad and Jan Mandel, *Spectra of sums of orthogonal projections and applications to parallel computing*, BIT **31** (1991), 76–88.
3. Marian Brezina and Petr Vaněk, *One black-box iterative solver*, UCD/CCM Report 106, Center for Computational Mathematics, University of Colorado at Denver, 1997, <http://www-math.cudenver.edu/ccmreports/rep106.ps.gz>.

4. Zhiqiang Cai, Thomas A. Manteuffel, and Stephen F McCormick, *First-order system least squares for second-order partial differential equations: Part ii.*, SIAM J. Numer. Anal. **34** (1997), 425–454.
5. Armel de La Bourdonnaye, *Une Méthode de discréétisation microlocale et son application à un problème de diffraction*, C. R. Acad. Sci. Paris, Serie I **318** (1994), 385–388.
6. Charles I. Goldstein, *Multigrid preconditioners applied to the iterative solution of singularly perturbed elliptic boundary value problems and scattering problems*, Innovative numerical methods in engineering, Proc. 4th Int. Symp., Atlanta/Ga., 1986 (Berlin) (R.P. Shaw, J. Periaux, A. Chaudouet, J. Wu, C. Marino, and C.A. Brebbia, eds.), Springer-Verlag, 1986, pp. 97–102.
7. W. Hackbusch, *A fast iterative method for solving Helmholtz's equation in a general region*, Fast Elliptic Solvers (U. Schumann, ed.), Advance Publications, London, 1978, pp. 112–124.
8. B. Lee, T. Manteuffel, S. McCormick, and J. Ruge, *Multilevel first-order system least squares (FOSLS) for Helmholtz equation*, Procs. 2nd International Conf. on Approx. and Num. Meths. for the Solution of the Maxwell Equations, Washington, D.C, John Wiley and Sons, 1993.
9. Jan Mandel, Marian Brezina, and Petr Vaněk, *Energy optimization of algebraic multigrid bases*, UCD/CCM Report 125, Center for Computational Mathematics, University of Colorado at Denver, February 1998, <http://www-math.cudenver.edu/ccmreports/rep125.ps.gz>.
10. Jan Mandel and Mirela Popa, *A multigrid method for elastic scattering*, UCD/CCM Report 109, Center for Computational Mathematics, University of Colorado at Denver, September 1997, <http://www-math.cudenver.edu/ccmreports/rep109.ps.gz>.
11. Petr Mayer and Jan Mandel, *The finite element ray method for the Helmholtz equation of scattering: First numerical experiments*, UCD/CCM Report 111, Center for Computational Mathematics, University of Colorado at Denver, October 1997.
12. B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, New York, 1996.
13. P. Vaněk, *Acceleration of convergence of a two level algorithm by smoothing transfer operators*, Appl. Math. **37** (1992), 265–274.
14. Petr Vaněk, Marian Brezina, and Jan Mandel, *Algebraic multigrid for problems with jumps in coefficients*, In preparation.
15. Petr Vaněk, Jan Mandel, and Marian Brezina, *Algebraic multigrid on unstructured meshes*, UCD/CCM Report 34, Center for Computational Mathematics, University of Colorado at Denver, December 1994, <http://www-math.cudenver.edu/ccmreports/rep34.ps.gz>.
16. ———, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing **56** (1996), 179–196.
17. Petr Vaněk, Jan Mandel, and Marian Brezina, *Solving a two-dimensional Helmholtz problem by algebraic multigrid*, UCD/CCM Report 110, Center for Computational Mathematics, University of Colorado at Denver, October 1997, <http://www-math.cudenver.edu/ccmreports/rep110.ps.gz>.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF COLORADO AT DENVER, DENVER, CO 80217-3364, AND UNIVERSITY OF WEST BOHEMIA, PLZEŇ, CZECH REPUBLIC
E-mail address: pvanek@math.cudenver.edu

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF COLORADO AT DENVER, DENVER, CO 80217-3364, AND CENTER FOR SPACE STRUCTURES, UNIVERSITY OF COLORADO AT BOULDER, BOULDER CO 80309-0429
E-mail address: jmandel@math.cudenver.edu

DEPARTMENT OF APPLIED MATHEMATICS, UNIVERSITY OF COLORADO AT BOULDER, BOULDER CO 80309-0526
E-mail address: mbrezina@math.cudenver.edu

A Comparison of Scalability of Different Parallel Iterative Methods for Shallow Water Equations

Arnt H. Veenstra, Hai Xiang Lin, and Edwin A.H. Vollebregt

1. Introduction

The scalability of the iterative methods Jacobi, SOR, CG and AMS-CG will be analyzed for the solution of systems of equations with symmetric positive definite matrices, arising from the shallow water equations. For the parallel solution of these systems domain decomposition is applied and each subdomain is computed by a processor. The AMS-CG (Approximate Multi-Subdomain CG) method is a relatively new CG-type method specially designed for parallel computation. We will show that it compares favorably in our experiments to the other methods with respect to speedup, communication overhead, convergence rate and scalability.

In order to predict the water movement and the dispersion of dissolved substances for the management of coastal seas and estuaries, computer simulations are performed which comprises the numerical solution of a.o. the shallow water equations. The shallow water equations are a set of coupled nonlinear partial differential equations which describe a 3-dimensional flow in shallow water [3]. The SWE are derived from the incompressible Navier Stokes equations by assuming a hydrostatic pressure distribution. In this way the pressure is eliminated and replaced by the unknown water level ζ . Practical simulation requires an accurate solution using a fine grid and a small time step. Thus, computation time becomes very important. In this paper we consider parallel solution in order to reduce computation time.

The partial differential equations are discretized using a staggered grid, see e.g. [7], and a two stage time splitting method [3]. This leads to a nonlinear pentadiagonal system of equations to be solved in each time step. These nonlinear systems are solved by linearization [3] and by using iterative methods for the resulting systems, which are symmetric positive definite. For the parallel solution of these systems domain decomposition is applied and each subdomain is computed by a separate processor. We analyze and evaluate the performance and scalability of different iterative methods for solving these systems: Jacobi, CG, SOR and AMS-CG [6].

The AMS-CG (Approximate Multi-Subdomain-CG) method is a relatively new CG-type method. The method is specially designed for (massively) parallel computation. Unlike CG, it only requires communication between neighboring subdomains and the unknowns in each subdomain are approximated with an independent

1991 *Mathematics Subject Classification*. Primary 65F10; Secondary 65Y05, 76B15.

search direction. It is known that the overhead of global communications, such as inner product computations in CG, can strongly affect the parallel performance and scalability on massively parallel processors. AMS-CG eliminates the global communication and therefore should have a better scalability.

We applied the following preconditioners for CG and AMS-CG: ILU [5], MILU [4] and FSAI [2]. Each of them is applied with overlapping subdomains and non-overlapping subdomains and with varying internal boundary conditions. We compare the above mentioned iterative methods combined with domain decomposition from the experimental results on a Cray T3D. The speedup, communication overhead, convergence rate, and scalability of the different methods are discussed.

2. Problem formulation

The application problem in our study is a 3-dimensional hydrodynamic sea model: A simplified form of the shallow water equations. These equations describe the motion and the elevation of the water:

$$(1) \quad \frac{\partial u}{\partial t} = fv - g \frac{\partial \zeta}{\partial x} + \frac{1}{h^2} \frac{\partial}{\partial \sigma} \left(\mu \frac{\partial u}{\partial \sigma} \right)$$

$$(2) \quad \frac{\partial v}{\partial t} = -fu - g \frac{\partial \zeta}{\partial y} + \frac{1}{h^2} \frac{\partial}{\partial \sigma} \left(\mu \frac{\partial v}{\partial \sigma} \right)$$

$$(3) \quad \frac{\partial \zeta}{\partial t} = -\frac{\partial}{\partial x} \left(h_0 \int^1 u d\sigma \right) - \frac{\partial}{\partial y} \left(h_0 \int^1 v d\sigma \right)$$

The equations have been transformed in the vertical direction into depth-following coordinates (σ) by the so-called sigma transformation: $\sigma = \frac{\zeta - z}{d + \zeta}$. The boundary conditions at the sea surface ($\sigma=0$) and at the bottom ($\sigma=1$) are:

$$(4) \quad \left(\mu \frac{\partial u}{\partial \sigma} \right)_{\sigma=0} = -\frac{h}{\rho} W_f \cos(\phi), \quad \left(v \frac{\partial v}{\partial \sigma} \right)_{\sigma=0} = -\frac{h}{\rho} W_f \sin(\phi)$$

$$(5) \quad \left(\mu \frac{\partial u}{\partial \sigma} \right)_{\sigma=1} = \frac{gu_d}{C^2}, \quad \left(v \frac{\partial v}{\partial \sigma} \right)_{\sigma=1} = \frac{gv_d}{C^2}$$

W =wind force, ϕ =direction of wind force and u_d and v_d velocities near the bottom. In the horizontal direction a staggered grid is used , see e.g. [7] for more information about the use and advantages of staggered grids. In the vertical direction the grid is divided into a few equidistant layers. The variable ζ is discretized as Z , which is the water level compared to a horizontal plane of reference. The spatial derivatives are discretized using second-order central differences. The time integration is performed by a two-stage time splitting method [3]. The time step is split in two stages: $t = n$ to $t = n + 1/2$ and $t = n + 1/2$ to $t = n + 1$. At the first stage the equations describing the vertical diffusion are treated implicitly. Each vertical line of grid points corresponds to a separate tridiagonal system of equations.

At the second stage the equations describing the propagation of the surface waves are treated implicitly. This pentadiagonal system of equations describes the water level related to the water level in 4 neighboring grid points:

$$(1 + c_1 + c_2 + c_3 + c_4)Z_{i,j}^{n+1} - c_1 Z_{i+1,j}^{n+1} - c_2 Z_{i-1,j}^{n+1} - c_3 Z_{i,j+1}^{n+1} - c_4 Z_{i,j-1}^{n+1} = Z_{i,j}^{n+1/2}$$

$Z_{i,j}^{n+1}$ is the water level in grid point (i,j) at time $t=n+1$. The coefficients c_i are functions depending on the water level Z . This equation can be put in the form:

$$A(Z^{n+1})Z^{n+1} = Z^{n+1/2}$$

The equations are first linearized before applying an iterative solution method by:

$$A(X^m)X^{m+1} = Z^{n+1/2}$$

Here $X^0 = Z^{n+1/2}$ and A is symmetric positive definite. These systems are linear in X^{m+1} . The iterative methods are used to compute X^{m+1} . Thus there are two iteration processes: The outer iteration process updates $A(X^m)$ and continues until $\|X^{m+1} - X^m\| < \varepsilon$ and the inner iteration process solves each X^{m+1} . Both processes are executed until convergence. In our experiments we compare the results of the iterative methods for the entire two iteration processes in the second stage.

3. Approximate Multi-Subdomain-CG

Starting with a basic form of CG without any optimizations, we can derive the Multi Subdomain CG (MS-CG) method by splitting the global search direction into local search directions, one for each subdomain. The local search direction p_d for subdomain d is a vector with zero entries for variables corresponding to positions outside subdomain d . Each iteration a p_d for each subdomain must be determined. The p_d 's are put in the columns of a matrix P . Instead of the scalars α and β in the CG method, α and β are vectors with a separate entry for each subdomain. We require that $p_d^k = z_d^{k-1} + \beta^k P^{k-1}$ is conjugate to all previous p_d^j with $j < k$. This is satisfied by computing β from $C^{k-1}\beta^k = -(Q^{k-1})^T z^{k-1}$. In order to minimize $\|x - x^k\|_A = (x - x^k, b - Ax^k)$, α is computed by $C^k\alpha^k = (P^k)^T r^{k-1}$, where $C^k = (Q^k)^T P^k$. It can be proven that the errors $\|x - x^k\|_A$ are non-increasing [6]. The preconditioned MS-CG method can be derived from PCG.

Algorithm: Multi-Subdomain CG
 x^0 = initial guess, $k = 0$, $r^0 = b - Ax^0$
 while 'not converged'
 solve $Mz_d^k = r_d^k$
 $k = k + 1$
 if ($k = 1$)
 for each subdomain d : $p_d^1 = z_d^0$
 else
 $C^{k-1}\beta^k = -(Q^{k-1})^T z^{k-1}$
 for each subdomain d : $p_d^k = z_d^{k-1} + \beta^k P^{k-1}$
 end
 $Q^k = AP^k$
 $C^k = (Q^k)^T P^k$
 $C^k\alpha^k = (P^k)^T r^{k-1}$
 $x^k = x^{k-1} + P^k\alpha^k$
 $r^k = r^{k-1} - Q^k\alpha^k$
 end

$$x = x^k$$

The matrix M has to be a symmetric positive definite preconditioning matrix. $p_d^i=0$ outside subdomain i . The matrix C is symmetric positive definite when the matrix A is. The CG method has the disadvantage that for each iteration 2 dot products are computed. Instead of these dot products the MS-CG method requires 2 matrix equations with the matrix C to be solved. The matrix C is symmetrical positive definite in our problem. The idea of the Approximate MS-CG (AMS-CG) method is to approximate the 2 equations with the matrix C by a few Jacobi iterations. The matrix C is relatively small: $l \times l$, where l = number of subdomains. The Jacobi method only requires local communication. The convergence of the Jacobi iterations is not tested, because we don't want to introduce global communication. Usually a small fixed number of Jacobi iterations (2 to 4) is sufficient. The AMS-CG method requires no global communication, except to check convergence. This convergence check can be done once after performing a number of iterations.

4. Preconditioning

In this section we describe several preconditioners for the CG-method and the AMS-CG method. ILU preconditioning [5] is well known. It is derived from the LU-decomposition. The LU-decomposition generates full matrices. The ILU(0) algorithm allows no fill in, ie: An entry in L and U is zero whenever the corresponding entry in the matrix A is zero.

Gustafsson [4] proposed the MILU (Modified ILU) method. The components which are disregarded in ILU are added to the main diagonal. Therefore the row sum of LU is equal to the row sum of A . In our experiments the MILU preconditioner performed much better than the ILU preconditioner. Therefore most experiments with preconditioning were done with MILU.

The ILU and MILU preconditioners have to be uncoupled in order to compute different subdomains in parallel. There are a few possibilities: We can simply disregard the coupling, we can impose boundary conditions on the internal boundaries or we can use a Schur complement method. We did not use a Schur complement method, however. This would have resulted in more communication, while we tried to precondition without communication or with very little communication. Consequently we found an increasing number of iterations for an increasing number of subdomains. With a Schur complement method the number of iterations will almost certainly be roughly the same for different numbers of subdomains.

The FSAI (Factorised Sparse Approximate Inverse) preconditioner [2] uses the following approximate inverse of A : $G^T D^{-1} G$, where $D=\text{diag}(G)$ and $g_{ij} = 0$ when $a_{ij} = 0$, $(GA)_{ij} = \delta_{ij}$ when $a_{ij} \neq 0$. This preconditioner can be applied to different numbers of subdomains without any problem. This only requires some local communication between neighboring processors. These preconditioners are constructed with non-overlapping or overlapping subdomains.

5. Numerical results

The test problem used is that of de Goede [3, p. 63]. We simulate a rectangular basin of $400 \text{ km} \times 800 \text{ km}$. The bottom is inclined in x -direction with a depth of 20 m at one end and a depth of 340 m at the other end. The grid used is 100×100 points with 5 vertical layers. Simulations are carried out with time steps equal to:

TABLE 1. Amount of computation per iteration

Method	Computations
Jacobi	$6n$ multiplications
SOR	$6n$ multiplications
CG	$10n$ multiplications
CG+MILU	$14n$ multiplications + $2n$ divisions
AMS-CG	$12n$ multiplications
AMS-CG+MILU	$16n$ multiplications + $2n$ divisions

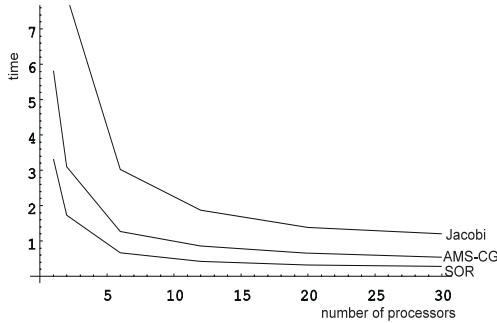


FIGURE 1. Average execution time per time step plotted against the number of processors

600 s, 1200 s and 2400s. We simulate a period of 4 hours and use a wind force of 1.0 N/m² in x -direction. For the AMS-CG method we take a number of 3 Jacobi iterations.

Convergence. A good indication of the convergence properties of an iterative method is the average number of iterations required per time step. The convergence properties of AMS-CG are just as good as the convergence properties of CG in our experiments. This holds with and without preconditioning. As expected the convergence properties of Jacobi are bad, especially for larger time steps. The convergence of the SOR method is better than the convergence of CG and AMS-CG when the right relaxation parameter is chosen. The convergence of the MILU-preconditioned CG and AMS-CG is best of all. To give an indication: With a time step of 1200 s and 6×5 subdomains the number of iterations for Jacobi is 1008, for CG/AMS-CG: 202, for SOR: 125, for CG+MILU/AMS-CG+MILU: 55.

Convergence of MILU-preconditioned CG or AMS-CG becomes worse as domain numbers increase. The largest difference is between 1 global domain (1×1) and 2 subdomains (2×1). We tried to improve this situation, but the number of iterations remained largely the same with overlapping or non-overlapping subdomains and with different internal boundary conditions. The convergence of FSAI-preconditioned AMS-CG is worse than the convergence of MILU-preconditioned AMS-CG. Therefore, in case of preconditioning, we will only present results with MILU using non-overlapping subdomains.

Execution time. For n grid points, the amount of computation per iteration for the different methods is shown in Table 1. CG and AMS-CG require more

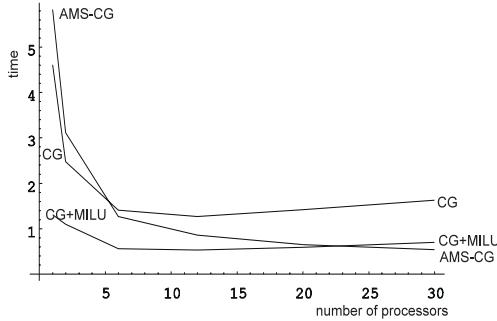


FIGURE 2. Average execution time per time step plotted against the number of processors

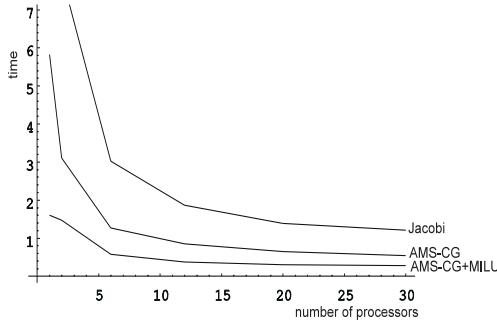


FIGURE 3. Average execution time per time step plotted against the number of processors

computations per iteration, with or without preconditioning. However, the number of iterations is less for CG and AMS-CG. This results in a better overall performance of CG and AMS-CG for a complete time step. Figure 1, 2 and 3 show the average execution time per time step plotted against the number of processors. Figure 1 shows that the average execution time of AMS-CG lies just between the execution time of Jacobi and SOR. Figure 2 shows that AMS-CG is worse than CG and CG+MILU on a small number of processors. If the number of processors increases AMS-CG becomes better than CG and CG+MILU. Figure 3 shows that the average execution time of AMS-CG+MILU is better than the execution time of AMS-CG. The execution time of AMS-CG+MILU is also better than the execution time of SOR.

Optimization. The convergence check implies that for every iteration a global dot product must be computed. Therefore it is advantageous not to check the convergence every iteration. In our experiments we choose to perform the convergence check only every 10 iterations. This reduces global communication considerably. For the standard CG method it is not relevant to do this, because one of the 2 dot products required per iteration is also used for the convergence check.

Speedup and scalability. Figure 4 shows the relative speedup compared to the execution time of the respective method on 2 processors with 2×1 subdomains. The figure is clear: The global communication of CG and CG+MILU results in a

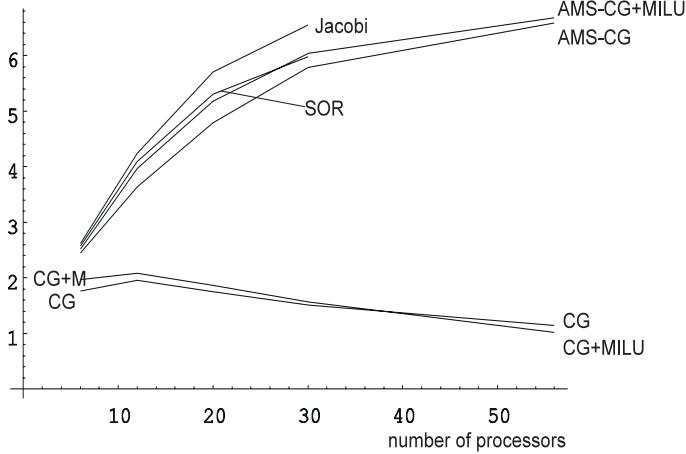


FIGURE 4. Relative speedup compared to the execution time on 2 processors (2×1 subdomains) of the respective method

TABLE 2. The ratio between the execution time on 30 processors (6×5 subdomains) and on 1 processor (1×1 subdomain)

method	time step 600 s	1200 s	2400 s
Jacobi	20.0	-	-
SOR	6.3	11.4	11.3
CG	16.3	16.6	16.9
CG+MILU	8.5	11.0	15.7
AMS-CG	10.0	9.9	8.4
AMS-CG+MILU	4.4	7.3	15.8

very bad speedup. A slowdown has been observed for larger number of processors. AMS-CG and AMS-CG+MILU have roughly the same speedup as Jacobi and SOR, which are known to be well parallelizable. For the evaluation of the scalability we kept the size of each subdomain constant. Each subdomain consisted of 50×50 grid points. The parallel execution times are compared with the execution time for a problem equal to 1 subdomain. Table 2 shows the execution time for 6×5 subdomains using 30 processors divided by the execution time for 1×1 subdomain using 1 processor. Note that the problem size grows proportionally with the number of subdomains. Thus, with the same number of processors, the execution time of 6×5 subdomains will be at least 30 times that of 1×1 subdomain. With perfect scalability, the ratio between the execution time for 6×5 subdomains using 30 processors and the execution time for 1×1 subdomain using 1 processor should be equal to 1. But in practice this ratio will be generally larger than 1, due to the increased number of iterations and the communication overhead.

The execution time of the Jacobi method with increasing number of subdomains grows fast. This is caused by the slow convergence of this method when the problem size increases. The SOR method does quite well in this case but its performance soon worsens as the time step increases (e.g. with an integration step of 1200 s the execution time ratio becomes 11.4). The CG and CG+MILU methods are not

good scalable because of the fast increase of the overhead of global communication. The AMS-CG+MILU has the best scalability for small integration time steps, but it becomes worse for large integration time steps. The reason for this is that the preconditioning is then less effective. The performance of the AMS-CG method is in between and its convergence is insensitive to the integration time step.

6. Conclusions

We compared the iterative methods Jacobi, SOR, CG and AMS-CG on the solution of a system of equations which describes wave propagation. The matrix associated with the system of equations is sparse (pentadiagonal). This results in few computations and relatively more communication. The (preconditioned) AMS-CG method is well suited for a parallel solution of this application problem in combination with domain decomposition. In this case the method has convergence properties comparable to (preconditioned) CG. The speedup of AMS-CG is much better than the speedup of CG. This is due to the fact that AMS-CG requires only local communication. MILU preconditioning results in better convergence.

It is remarkable that the number of iterations of the preconditioned AMS-CG makes a big jump from 1 subdomain to 2 subdomains, while this is not observed for AMS-CG. The number of iterations increases only slightly from 2 subdomains to 3 subdomains and more. Future research should analyze the preconditioner and possibilities to improve it. An interesting topic is to implement a Schur complement preconditioner and compare the convergence and scalability. The Schur complement preconditioner generally has a faster convergence [1], but it requires more communication.

Also interesting is to apply the AMS-CG method to other type of problems. It is still an open question if the AMS-CG method is a suitable parallel method for the solution of systems of equations arising from other applications.

References

1. T.F. Chan and D. Goovaerts, *A note on the efficiency of domain decomposed incomplete factorizations*, SIAM J. Sci. Stat. Comput. **11** (1990), no. 4, 794–802.
2. M. R. Field, *An efficient parallel preconditioner for the conjugate gradient algorithm*, presented at the IMACS 97 conference, Jackson Hole, Wyoming, USA, july 1997.
3. E. de Goede, *Numerical methods for the three-dimensional shallow water equations on supercomputers*, Ph.D. thesis, University of Amsterdam, The Netherlands, 1992.
4. I. Gustafsson, Preconditioning Methods. Theory and Applications (David J. Evans, ed.), ch. Modified Incomplete Choleski Methods, pp. 265–293, Gordon and Breach, Science Publishers, New York, London, Paris, 1983, pp. 265–293.
5. J.A. Meijerink and H.A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. **31** (1977), 148–162.
6. E.A.H. Vollebregt, *Multi-subdomain conjugate gradients: A global inner product free cg-type method*, Tech. report, Department of Applied Math.&Informatics, TU Delft, 1997.
7. C.B. Vreugdenhil, *Numerical methods for shallow-water flow*, Water Science and Technology Library, vol. 13, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1994.

FACULTY OF INFORMATION TECHNOLOGY AND SYSTEMS, DELFT UNIVERSITY OF TECHNOLOGY,
MEKELWEG 4, 2628 CD, DELFT, THE NETHERLANDS
E-mail address: h.x.lin@math.tudelft.nl

A Nonoverlapping Subdomain Algorithm with Lagrange Multipliers and its Object Oriented Implementation for Interface Problems

Daoqi Yang

1. Introduction

A parallel nonoverlapping subdomain Schwarz alternating algorithm with Lagrange multipliers and interface relaxation is proposed for linear elliptic interface problems with discontinuities in the solution, its derivatives, and the coefficients. New features of the algorithm include that Lagrange multipliers are introduced on the interface and that it is used to solve equations with discontinuous solution. These equations have important applications to alloy solidification problems [1] and immiscible flow of fluids with different densities and viscosities and surface tension. They do not fit into the Schwarz preconditioning and Schur complement frameworks since the solution is not in $H^1(\Omega)$, but is piecewise in $H^1(\Omega)$, where Ω is the physical domain on which the differential equations are defined. An expression for the optimal interface relaxation parameters is also given. Numerical experiments are conducted for a piecewise linear triangular finite element discretization in object oriented paradigm using C++. In this implementation, the class for subdomain solvers inherits from the class for grid triangulation and contains type bound procedures for forming stiffness matrices and solving linear systems. From software engineering point of view, features like encapsulation, inheritance, polymorphism and dynamic binding, make such domain decomposition algorithms an ideal application area of object oriented programming.

The organization of this work is as follows. In Section 2, the domain decomposition method is described for general elliptic interface problems. In Section 3, a finite element approximation with Lagrange multipliers is considered. Then in Section 4, some implementation issues using object oriented techniques in C++ are discussed. Finally in Section 5, numerical examples are provided to check the performance of the method.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N30.

Key words and phrases. Interface Problems, Lagrange Multipliers, Interface Relaxation, Object Oriented Programming.

This work was supported in part by Institute for Mathematics and its Applications at University of Minnesota. No version of this paper will be published elsewhere.

2. The Domain Decomposition Method

Let Ω be a smooth bounded two-dimensional domain or a convex polygon with boundary $\partial\Omega$. Assume that Ω is the union of two nonoverlapping subdomains Ω_1 and Ω_2 with interface Γ ; that is, $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$, $\Omega_1 \cap \Omega_2 = 0$, and $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. When Ω_1 and Ω_2 are disconnected, the decomposition can actually contain more than two subdomains.

Consider the following elliptic interface problem: find $u_1 \in H^1(\Omega_1)$ and $u_2 \in H^1(\Omega_2)$ such that

$$\begin{aligned} (1) \quad L_k u_k &= f \text{ in } \Omega_k, \quad k = 1, 2, \\ (2) \quad u_k &= g \text{ on } \partial\Omega_k \cap \partial\Omega, \quad k = 1, 2, \\ (3) \quad u_1 - u_2 &= \mu \text{ on } \Gamma, \\ (4) \quad \frac{\partial u_1}{\partial \nu_A^1} + \frac{\partial u_2}{\partial \nu_A^2} &= \eta \text{ on } \Gamma, \end{aligned}$$

where for $k = 1, 2$,

$$L_k u = - \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} \left(a_{ij}^{(k)}(x) \frac{\partial u}{\partial x_j} \right) + b^{(k)}(x)u, \quad \frac{\partial u}{\partial \nu_A^k} = \sum_{i,j=1}^2 a_{ij}^{(k)} \frac{\partial u}{\partial x_j} \nu_i^k,$$

$\nu^k = \{\nu_1^k, \nu_2^k\}$ is the outward normal unit vector to $\partial\Omega_k$. I assume that $f \in L^2(\Omega)$ and $g \in H^{1/2}(\partial\Omega)$, and that $\mu(x)$ and $\eta(x)$ are given regular functions on Γ . That is, the solution of (1)-(4) is sought with specified strength of discontinuity, and so is its conormal derivative. I also assume that the coefficients $\{a_{ij}^{(k)}\}$ are symmetric, uniformly positive definite, and bounded on Ω_k , and $b^{(k)} \geq 0$ on Ω_k .

Equations of type (1)-(4) need be solved at each time step for the two-phase generalized Stephan problems as encountered in alloy solidification processes [1, pages 8, 15]. They have applications in many wave propagation and fluid flow problems [1, 5]. In some cases, the interface conditions (3) and (4) are nonlinear [1].

Schwarz overlapping and substructuring domain decomposition methods have been analyzed for the special linear case (1)-(4) in which $\mu = 0$ and $\eta = 0$. It is not clear that any of these methods applies directly to the general case with $\mu \neq 0$ and $\eta \neq 0$, since the solution is not in the space $H^1(\Omega)$, although its restriction to Ω_k is in the space $H^1(\Omega_k)$. Finite element methods on the whole domain Ω without domain decomposition do not seem to apply. In [5], a special kind of finite difference method without domain decomposition was considered for the problem (1)-(4).

I now define formally the following domain decomposition method for problem (1)-(4): Choose $u_k^0 \in H^1(\Omega_k)$ satisfying $u_k^0|_{\Omega \cap \Omega_k} = g$, $k = 1, 2, \dots$, For $n = 0, 1, 2, \dots$,

the sequence $u_k^n \in H^1(\Omega_k)$ with $u_k^n|_{\partial\Omega \cap \partial\Omega_k} = g$ is constructed such that

$$(5) \quad \begin{cases} L_1 u_1^{2n+1} = f \text{ in } \Omega_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1-\alpha)u_2^{2n} + (1-\alpha)\mu \text{ on } \Gamma; \end{cases}$$

$$(6) \quad \begin{cases} L_2 u_2^{2n+1} = f \text{ in } \Omega_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1-\alpha)u_2^{2n} - \alpha\mu \text{ on } \Gamma; \end{cases}$$

$$(7) \quad \begin{cases} L_1 u_1^{2n+2} = f \text{ in } \Omega_1, \\ \frac{\partial u_1^{2n+2}}{\partial \nu_A^1} = \beta \frac{\partial u_1^{2n+1}}{\partial \nu_A^1} + (1-\beta) \frac{\partial u_2^{2n+1}}{\partial \nu_A^1} + (1-\beta)\eta \text{ on } \Gamma; \end{cases}$$

$$(8) \quad \begin{cases} L_2 u_2^{2n+2} = f \text{ in } \Omega_2, \\ \frac{\partial u_2^{2n+2}}{\partial \nu_A^2} = \beta \frac{\partial u_1^{2n+1}}{\partial \nu_A^2} + (1-\beta) \frac{\partial u_2^{2n+1}}{\partial \nu_A^2} + \beta\eta \text{ on } \Gamma; \end{cases}$$

where $\alpha, \beta \in (0, 1)$ are relaxation parameters that will be determined to ensure and accelerate the convergence of the iterative procedure. This algorithm is motivated by the ones proposed by Funaro, Quarteroni, and Zanolli [4], Marini and Quarteroni [7, 8], Lions [6], Després [2], Rice, Vavalis, and Yang [10], Quarteroni [9], Douglas and Yang [3], and Yang [11, 12], where regular problems with $\mu = 0$ and $\eta = 0$ were considered.

This algorithm is different from others in that Dirichlet and Neumann subdomain problems were solved at the same iteration levels but on different subdomains in [4, 7, 8, 11] and Robin subdomain problems were solved in [6, 2, 12]. For problems with continuous solution and coefficients, this algorithm reduces to [10, 3], where convergence results for general coefficients were not provided. In [13], a detailed analysis for the algorithm will be made at the differential and discrete levels and numerical tests using finite difference methods will be conducted.

3. Finite Element Approximation with Lagrange Multipliers

In this section, I reformulate the algorithm (5)-(8) by introducing Lagrange multipliers on the interface to replace conormal derivatives. Finite element discretization is then applied to subdomain problems. Although Lagrange multipliers have been used in other contexts, it does not appear that they have been employed to (5)-(8) before. I first introduce some notation. Denote the Hilbert spaces

$$V_k = \{v \in H^1(\Omega_k), \quad v|_{\partial\Omega \cap \partial\Omega_k} = 0\}, \quad k = 1, 2,$$

and let γ_0 be the trace operator from V_k onto $H_0^{1/2}(\Gamma)$. Define the bilinear forms:

$$(9) \quad a_k(u, w) = \sum_{i,j=1}^2 \int_{\Omega_k} a_{ij}^{(k)} \frac{\partial u}{\partial x_j} \frac{\partial w}{\partial x_i} dx + \int_{\Omega_k} b^{(k)} uw dx, \quad k = 1, 2,$$

$$(10) \quad (u, w)_k = \int_{\Omega_k} uw dx, \quad k = 1, 2.$$

For convenience I use the following norms in V_1 and V_2 ,

$$(11) \quad \|w\|_k^2 = a_k(w, w), \quad \forall w \in V_k, \quad k = 1, 2.$$

Then the variational formulation of the scheme (5)-(8) can be written as:

$$(12) \quad \begin{cases} a_1(u_1^{2n+1}, w) - \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^1}, \gamma_0 w \right\rangle = (f, w)_1, & \forall w \in V_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} + (1 - \alpha) \mu \text{ on } \Gamma; \end{cases}$$

$$(13) \quad \begin{cases} a_2(u_2^{2n+1}, w) - \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^2}, \gamma_0 w \right\rangle = (f, w)_2, & \forall w \in V_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} - \alpha \mu \text{ on } \Gamma; \end{cases}$$

$$(14) \quad a_1(u_1^{2n+2}, w) = \beta \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^1}, \gamma_0 w \right\rangle + (1 - \beta) \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^1} + \eta, \gamma_0 w \right\rangle + (f, w)_1, \quad \forall w \in V_1;$$

$$(15) \quad a_2(u_2^{2n+2}, w) = \beta \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^2} + \eta, \gamma_0 w \right\rangle + (1 - \beta) \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^2}, \gamma_0 w \right\rangle + (f, w)_2, \quad \forall w \in V_2;$$

for $k = 1, 2$. Here $\langle \cdot, \cdot \rangle$ denotes the inner product over the interface Γ , or the duality between $H_{00}^{1/2}(\Gamma)$ and its dual space.

Replacing the conormal derivatives $\frac{\partial u_k^n}{\partial \nu_A^k}$ by the Lagrange multipliers λ_k^n , (12)-(15) becomes

$$(16) \quad \begin{cases} a_1(u_1^{2n+1}, w) - \langle \lambda_1^{2n+1}, \gamma_0 w \rangle = (f, w)_1, & \forall w \in V_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} + (1 - \alpha) \mu \text{ on } \Gamma; \end{cases}$$

$$(17) \quad \begin{cases} a_2(u_2^{2n+1}, w) - \langle \lambda_2^{2n+1}, \gamma_0 w \rangle = (f, w)_2, & \forall w \in V_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} - \alpha \mu \text{ on } \Gamma; \end{cases}$$

$$(18) \quad \begin{aligned} a_1(u_1^{2n+2}, w) &= \langle \beta \lambda_1^{2n+1} - (1 - \beta) \lambda_2^{2n+1} + (1 - \beta) \eta, \gamma_0 w \rangle + (f, w)_1, \quad \forall w \in V_1; \\ a_2(u_2^{2n+2}, w) &= \langle -\beta \lambda_1^{2n+1} + (1 - \beta) \lambda_2^{2n+1} + \beta \eta, \gamma_0 w \rangle + (f, w)_2, \quad \forall w \in V_2. \end{aligned}$$

The procedure (16)-(19) is the domain decomposition method with Lagrange multipliers at the differential level, a variant of (5)-(8). I now formulate its finite element version. Let $T_h = \{T\}$ be a regular triangulation of Ω with no elements crossing the interface Γ . I define finite element spaces, for $k = 1, 2$,

$$(20) \quad W_k^h = \{w \in H^1(\Omega_k) : w|_T \in P_r(T) \quad \forall T \in T_h, w|_{\partial\Omega \cap \partial\Omega_k} = 0\},$$

where $P_r(T)$ denotes the space of polynomials of degree $\leq r$ on T . Let Z^h be the space of the restrictions on the interface of the functions in W_k^h . Note that there are two copies of such a space assigned on Γ , one from Ω_1 and the other from Ω_2 . I denote them by Z_1^h and Z_2^h , respectively. Let $\{U_k^n, \Lambda_k^n\} \in W_k^h \times Z_k^h$ denote the finite element approximation of $\{u_k^n, \lambda_k^n\}$. Then, the finite element domain decomposition

method with Lagrange multipliers is constructed as follows:

$$(21) \quad \begin{cases} a_1(U_1^{2n+1}, w) - \langle \Lambda_1^{2n+1}, \gamma_0 w \rangle = (f, w)_1, & \forall w \in W_1^h, \\ U_1^{2n+1} = \alpha U_1^{2n} + (1 - \alpha) U_2^{2n} + (1 - \alpha)\mu \text{ on } \Gamma; \end{cases}$$

$$(22) \quad \begin{cases} a_2(U_2^{2n+1}, w) - \langle \Lambda_2^{2n+1}, \gamma_0 w \rangle = (f, w)_2, & \forall w \in W_2^h, \\ U_2^{2n+1} = \alpha U_1^{2n} + (1 - \alpha) U_2^{2n} - \alpha\mu \text{ on } \Gamma; \end{cases}$$

$$(23) \quad \begin{aligned} a_1(U_1^{2n+2}, w) \\ = \langle \beta \Lambda_1^{2n+1} - (1 - \beta) \Lambda_2^{2n+1} + (1 - \beta)\eta, \gamma_0 w \rangle + (f, w)_1, \quad \forall w \in W_1^h; \end{aligned}$$

$$(24) \quad \begin{aligned} a_2(U_2^{2n+2}, w) \\ = \langle -\beta \Lambda_1^{2n+1} + (1 - \beta) \Lambda_2^{2n+1} + \beta\eta, \gamma_0 w \rangle + (f, w)_2, \quad \forall w \in W_2^h. \end{aligned}$$

In order to give a convergence result, I introduce two linear operators. Let $\Phi^h = \{w|_\Gamma : w \in W_k^h, k = 1, 2\}$. Define the extension operators $R_k : \phi \in \Phi^h \rightarrow \{R_k^1 \phi, R_k^2 \phi\} \in W_k^h \times Z_k^h$ by

$$(25) \quad a_k(R_k^1 \phi, w) - \langle R_k^2 \phi, w \rangle = 0, \quad \forall w \in W_k^h, \quad R_k^1 \phi = \phi \text{ on } \Gamma.$$

Define $\bar{\sigma}$ and $\bar{\tau}$ to be two smallest finite real numbers such that

$$(26) \quad \sup_{\phi \in \Phi^h} \frac{\|R_1^1 \phi\|_2^2}{\|R_2^1 \phi\|_2^2} \leq \bar{\sigma}, \quad \sup_{\phi \in \Phi^h} \frac{\|R_2^1 \phi\|_2^2}{\|R_1^1 \phi\|_2^2} \leq \bar{\tau}.$$

The following results will be proved in [13].

THEOREM 1. *The domain decomposition method (21)-(24) is convergent in the energy norm if $\max\{0, 1 - \frac{2(\bar{\tau}+1)}{\bar{\tau}\bar{\sigma}^2+\bar{\tau}+2}\} < \alpha < 1$ and $\max\{0, 1 - \frac{2(\bar{\sigma}+1)}{\bar{\tau}^2\bar{\sigma}+\bar{\sigma}+2}\} < \beta < 1$. The optimal relaxation parameters are $\alpha = \frac{\bar{\sigma}^2\bar{\tau}+1}{\bar{\tau}\bar{\sigma}^2+\bar{\tau}+2}$ and $\beta = \frac{\bar{\sigma}\bar{\tau}^2+1}{\bar{\tau}^2\bar{\sigma}+\bar{\sigma}+2}$. Furthermore, the convergence is independent of the grid size h .*

4. Object Oriented Implementation

The finite element domain decomposition algorithm (21)-(24) can be implemented in an elegant way using the object oriented programming paradigm. Features such as information hiding, data encapsulation, inheritance, dynamic binding, and operator overloading can be employed very nicely in the algorithm. Below I give a brief description of my implementation details in the terminology of C++.

I first define a base class called “grid”, which generates a triangular grid in a subdomain. Data members like vertices of the triangles and coordinates of the vertices are protected members which can only be accessed by its members and derived class. Function members like getting the total degrees of freedom and printing a Matlab file for representing the triangulation are public that can be accessed by any program. Each object of the class grid represents the triangulation on each subdomain. The grid on different subdomains may not have to match on the interface. However, for simplicity, I apply matching grids on the subdomains.

Then I define a derived class called “subdomain”, which inherits from the class grid. The class subdomain contains private member functions for forming the stiffness matrix and for performing the Dirichlet and Neumann sweeps. The stiffness matrix forming function can access the triangulation information like triangle vertices and coordinates of vertices, which are protected members of the base class grid. In this step, I make use of my linear algebra library (which is built on object oriented programming for matrix and vector manipulations) for numerical integration. A numerical quadrature can be viewed as an inner product of two vectors;

using operator overloading, vectors can be multiplied directly, instead of an explicit function call like in Fortran or C. The Dirichlet sweep member function first solves the subdomain problem with Dirichlet boundary condition by choosing basis functions vanishing on the interface and then finds the Lagrange multipliers by choosing basis functions vanishing in the interior of the subdomain. The Neumann sweep member function just solves the subdomain problem with Neumann boundary condition on the interface. From these two steps, we see that the finite element method with Lagrange multipliers on the interface is easier to implement than finite difference or finite element methods without Lagrange multipliers [5, 13]. In particular, the subdomain finite dimensional problems are symmetric and positive definite in our case, as opposed to unsymmetric and indefinite problems in [5].

Finally, a friend to the class subdomain is implemented to coordinate the Dirichlet and Neumann sweeps and check the stopping criterion for the iterative process. This friend function takes an array of subdomain objects as arguments and has access to protected members of class grid and all members of class subdomain.

At the linear system solving steps inside the Dirichlet and Neumann sweeps, I first define an abstract matrix class that just contains the number of rows of the matrix, two pure virtual functions for matrix-vector multiplication and preconditioning, and a function for the preconditioned conjugate gradient method. Since the preconditioned conjugate gradient algorithm can be implemented once we have a matrix-vector multiplication function and a preconditioning function, it is defined in the abstract matrix class and inherited by classes for banded matrices and sparse matrices. With operator overloading (one kind of polymorphism), the preconditioned conjugate gradient function can be written in about the same number of lines and format as the algorithm (which increases the readability of the code) and is defined only once in the abstract class. It then can be called in the derived classes for banded matrices and sparse matrices which need to define the matrix-vector multiplication and preconditioning functions according to their data structures of the matrix storage. However, the banded Gauss elimination function has to be defined in every derived class since it can not be performed without knowing the structure of the matrix. Note that the banded matrix inherits the row number from the abstract matrix class and needs to define a data member for the bandwidth. For unsymmetric problems, GMRES instead of conjugate gradient method or banded Gauss elimination is used.

5. Numerical Examples

In this section, I present some numerical experiments for the iterative procedure (21)-(24). In all of my test, I let the domain $\Omega = \{(x, y) : 0 < x < 1, 0 < y < 0.5, x \geq y\} \cup \{(x, y) : 0 < x < 0.5, 0 < y < 1, x \leq y\}$ and the interface Γ be the line segment $\{(x, y) : x = y, 0 \leq x \leq 0.5\}$, which divides Ω into $\Omega_1 = \{(x, y) : 0 < x < 1, 0 < y < 0.5, x > y\}$ and $\Omega_2 = \{(x, y) : 0 < x < 0.5, 0 < y < 1, x < y\}$. Piecewise linear triangular finite elements are applied on each subdomain. One copy of the solution at the interface is kept from each subdomain due to its discontinuity. Note that a global finite element solution on the whole domain Ω may not exist. The relaxation parameters α and β are taken to be 0.5, which leads to fast convergence and thus optimal parameters are not used. Initial guess is chosen to be zero in all cases. I define iterative errors as the relative errors between the iterates at the current and previous iteration levels and true errors as the relative errors between

TABLE 1. Iterative and true errors for Example 2. The errors are shown in the L^∞ -norm.

Iteration	Grid size $\frac{1}{40} \times \frac{1}{40}$		Grid size $\frac{1}{80} \times \frac{1}{80}$	
	Iterative error	True error	Iterative error	True error
1	3.16E-1	2.75E-1	3.25E-1	2.93E-1
2	3.77E-2	1.05E-2	3.96E-2	3.21E-2
3	4.97E-3	5.33E-3	5.28E-3	7.55E-3
4	6.78E-4	6.01E-3	7.49E-4	2.27E-3
5	9.76E-5	5.91E-3	1.13E-4	3.02E-3

the current iterate and the true solution. All norms will be measured in the discrete $\max(\|\cdot\|_{L^\infty(\Omega_1)}, \|\cdot\|_{L^\infty(\Omega_2)})$ sense, which is different from the discrete $L^\infty(\Omega)$ norm since there are two different values on the interface from the two subdomains due to discontinuity.

EXAMPLE 2. Let

$$\begin{aligned} -\frac{\partial}{\partial x}(e^x \frac{\partial u_1}{\partial x}) - \frac{\partial}{\partial y}(e^y \frac{\partial u_1}{\partial y}) + \frac{1}{1+x+y} u_1 &= f_1, \quad \text{in } \Omega_1, \\ -\frac{\partial^2 u_2}{\partial x^2} - \frac{\partial^2 u_2}{\partial y^2} &= f_2, \quad \text{in } \Omega_2, \\ u_k = g_k, \quad \text{on } \partial\Omega_k \cap \partial\Omega, k &= 1, 2. \end{aligned}$$

The functions f_1 , f_2 , g_1 , g_2 , μ , and η are chosen such that the exact solution is

$$u_1(x, y) = 10x + y, \quad \text{in } \Omega_1, \quad u_2(x, y) = \sin(x + y), \quad \text{in } \Omega_2.$$

Table 1 shows the results for the iterative and true errors in the maximum norm on the whole domain Ω .

Now we consider a more difficult problem with convection on one side of the interface and general variable coefficients. In the Stephan problem [1], for the case of two incompressible phases, one solid and one liquid, with different densities, a convective term must be added to the heat-flow equation in the liquid region in order for mass to be conserved across phase-change interface.

EXAMPLE 3. Let

$$\begin{aligned} -\nabla \cdot \left(\begin{bmatrix} e^x & x \\ x & e^y \end{bmatrix} \nabla u_1 \right) + \left[\frac{1}{1+x+y}, \frac{1}{1+x+y} \right] \nabla u_1 + \frac{u_1}{1+x+y} \\ = f_1(x, y), \quad \text{in } \Omega_1, \\ -\nabla \cdot \left(\begin{bmatrix} x+1 & \sin(xy) \\ \sin(xy) & y+1 \end{bmatrix} \nabla u_2 \right) + (2 + \sin(x) + \cos(y))u_2 \\ = f_2(x, y), \quad \text{in } \Omega_2, \\ u_k = g_k, \quad \text{on } \partial\Omega_k \cap \partial\Omega, k = 1, 2. \end{aligned}$$

The functions f_1 , f_2 , g_1 , g_2 , μ , and η are chosen such that the exact solution is

$$u_1(x, y) = e^{xy}, \quad \text{in } \Omega_1, \quad u_2(x, y) = 5e^{xy} \sin(13x) \cos(13y), \quad \text{in } \Omega_2.$$

Table 2 shows the results for the iterative and true errors in the maximum norm on the whole domain Ω .

Numerical experiments show that the iterative method is insensitive to strong discontinuities in the solution and coefficients and leads to accurate approximate

TABLE 2. Iterative and true errors for Example 3. The errors are shown in the L^∞ -norm.

Iteration	Grid size $\frac{1}{40} \times \frac{1}{40}$		Grid size $\frac{1}{80} \times \frac{1}{80}$	
	Iterative error	True error	Iterative error	True error
1	1.97E-1	3.84E-1	2.33E-1	5.02E-1
2	8.77E-3	5.82E-3	1.09E-2	9.56E-3
3	3.38E-5	3.21E-3	4.16E-5	1.53E-3
4	1.41E-7	3.18E-3	1.78E-7	1.48E-3
5	6.27E-10	3.18E-3	8.29E-10	1.48E-3

solutions. Although the relaxation parameters α and β can be chosen in some optimal fashion, the method converges pretty fast with $\alpha = \beta = 1/2$ even for very complicated problems. It is observed that this method converges faster than Lions type methods [6] with a few subdomains which is suitable for most interface problems. Also, sharp interfaces of the true solution can be captured fairly easily and accurately. In [13], finite difference methods are applied to subdomain problems and similar numerical results are obtained.

To my knowledge, this paper is the first one to apply the Schwarz domain decomposition methodology to interface problems with discontinuous solution, conormal derivatives, and coefficients. This is a first attempt to solve time-dependent generalized Stephan problems such as alloy solidification and immiscible flow with surface tension. Applying non-matching grids will make my algorithm more attractive and suitable for such problems in that finite element grids can be generated separately in different subdomains and collaborative PDE solvers can be applied. Another salient feature of this algorithm is that symmetrical and positive definite linear systems are solved at each iteration for problems like Example 2 which is not symmetrical and positive definite in the whole domain. Implementations in the object oriented paradigm also make the algorithm easier to code and modify to meet the need of more complicated situations. Our future work will try to solve two and three dimensional application problems with more complex geometry and interface. Mortar elements may also be applied to this kind of problems.

References

1. J. Crank, *Free and moving boundary problems*, Oxford University Press, Oxford, 1984.
2. B. Despres, *Domain decomposition method and the Helmholtz problems*, Mathematical and Numerical Aspects of Wave Propagation Phenomena (G. Cohen, L. Halpern, and P. Joly, eds.), SIAM, Philadelphia, 1991, pp. 44–52.
3. J. Douglas, Jr. and D. Q. Yang, *Numerical experiments of a nonoverlapping domain decomposition method for partial differential equations*, Numerical Analysis: A. R. Mitchell 75th Birthday Volume (D. Griffiths and G. A. Watson, eds.), World Scientific, Singapore, 1996, pp. 85–97.
4. D. Funaro, A. Quarteroni, and P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal. **25** (1988), 1213–1236.
5. R. J. LeVeque and Z. L. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal. **31** (1994), 1019–1044.
6. P.-L. Lions, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1990, pp. 202–223.

7. L. D. Marini and A. Quarteroni, *An iterative procedure for domain decomposition methods: a finite element approach*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Péliaux, eds.), SIAM, 1988, pp. 129–143.
8. ———, *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math. **55** (1989), 575–598.
9. A. Quarteroni, *Domain decomposition methods for wave propagation problems*, Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering (D. E. Keyes *et al.*, ed.), SIAM, Philadelphia, 1995, pp. 21–38.
10. J. R. Rice, E. A. Vavalis, and D. Q. Yang, *Convergence analysis of a nonoverlapping domain decomposition method for elliptic PDEs*, J. Comput. Appl. Math. **87** (1997), 11–19.
11. D. Q. Yang, *A parallel iterative nonoverlapping domain decomposition procedure for elliptic problems*, IMA J. Numer. Anal. **16** (1996), 75–91.
12. ———, *A parallel nonoverlapping Schwarz domain decomposition algorithm for elliptic partial differential equations*, Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing (Philadelphia) (M. Heath *et al.*, ed.), SIAM, 1997.
13. ———, *A parallel nonoverlapping Schwarz domain decomposition method for elliptic interface problems*, (In preparation).

DEPARTMENT OF MATHEMATICS, WAYNE STATE UNIVERSITY, DETROIT, MI 48202-3483, USA.
E-mail address: dyang@na-net.ornl.gov, <http://www.math.wayne.edu/~yang>

Part 3

Theory

A Robin-Robin Preconditioner for an Advection-Diffusion Problem

Yves Achdou and Frédéric Nataf

1. Introduction

We propose a generalization of the Neumann-Neumann preconditioner for the Schur domain decomposition method applied to a advection diffusion equation. Solving the preconditioner system consists of solving boundary value problems in the subdomains with suitable Robin conditions, instead of Neumann problems. Preliminary tests assess the good behavior of the preconditioner.

The Neumann-Neumann preconditioner is used for the Schur domain decomposition method applied to symmetric operators, [5]. The goal of this paper is to propose its generalization to non symmetric operators. We replace the Neumann boundary conditions by suitable Robin boundary conditions which take into account the non symmetry of the operator. The choice of these conditions comes from a Fourier analysis, which is given in Sec. 2. When the operator is symmetric the proposed Robin boundary conditions reduce to Neumann boundary conditions. Also as in the symmetric case, the proposed preconditioner is exact for two subdomains and a uniform velocity. The preconditioner is presented in the case of a domain decomposed into non overlapping strips: the case of a more general domain decomposition will be treated in a forthcoming work as well as the addition of a coarse space solver.

The paper is organized as follows. In Sec. 2, the method is defined at the continuous level. In Sec. 3, the proposed preconditioner is constructed directly at the algebraic level. This may be important, if the grid is coarse and if upwind methods are used because the preconditioner defined at the continuous level is not relevant. In Sec. 4, we propose an extension to the case of nonmatching meshes (mortar method) [3] [1]. In Sec. 5, numerical results are shown for both conforming and nonconforming domain decompositions (mortar method).

2. The Continuous Case

We consider an advection-diffusion equation

$$\begin{aligned}\mathcal{L}(u) &= cu + \vec{a} \cdot \nabla u - \nu \Delta u = f && \text{in } \Omega =]0, L[\times]0, \eta[, \\ u &= 0 && \text{on } \partial\Omega.\end{aligned}$$

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 76D30.

The positive constant c may arise from a time discretization by an Euler implicit scheme of the time dependent equation. The equation is solved by a primal Schur method. We focus on the case when the domain is decomposed into non overlapping vertical strips $\Omega_k =]l_k, l_{k+1}[\times]0, \eta[$, $1 \leq k \leq N$. Let $\Gamma_{k,k+1} = \{l_{k+1}\} \times]0, \eta[$.

REMARK 1. The general case of an arbitrary domain decomposition will be treated in a forthcoming work.

We introduce

$$\begin{aligned} \mathcal{S} : (H_{00}^{1/2}(]0, \eta[))^{N-1} \times L^2(\Omega) &\rightarrow (H^{-1/2}(]0, \eta[))^{N-1} \\ ((u_k)_{1 \leq k \leq N-1}, f) &\mapsto \left(\frac{1}{2} \nu \left(\frac{\partial v_k}{\partial n_k} + \frac{\partial v_{k+1}}{\partial n_{k+1}} \right) \right)_{1 \leq k \leq N-1} \end{aligned}$$

where v_k satisfies

- (1) $\mathcal{L}(v_k) = f$ in Ω_k ,
- (2) $v_k = u_k$ on $\Gamma_{k,k+1}$ for $1 \leq k \leq N-1$,
- (3) $v_k = u_{k-1}$ on $\Gamma_{k-1,k}$ for $2 \leq k \leq N$,
- (4) $v_k = 0$ on $\partial\Omega \cap \partial\Omega_k$.

It is clear that $U = (u_{\Gamma_{k,k+1}})_{1 \leq k \leq N-1}$ satisfies

$$\mathcal{S}(U, 0) = -\mathcal{S}(0, f).$$

At the continuous level, we propose an approximate inverse of $\mathcal{S}(., 0)$ defined by

$$\begin{aligned} \mathcal{T} : (H^{-1/2}(]0, \eta[))^{N-1} &\rightarrow (H_{00}^{1/2}(]0, \eta[))^{N-1}, \\ (g_k)_{1 \leq k \leq N-1} &\mapsto \left(\frac{1}{2} (v_k + v_{k+1})_{\Gamma_{k,k+1}} \right)_{1 \leq k \leq N-1}, \end{aligned}$$

where v_k satisfies

- (5) $\mathcal{L}(v_k) = 0$ in Ω_k ,
- (6) $(\nu \frac{\partial}{\partial n_k} - \frac{\vec{a} \cdot \vec{n}_k}{2}) (v_k) = g_k$ on $\Gamma_{k,k+1}$ for $1 \leq k \leq N-1$,
- (7) $(\nu \frac{\partial}{\partial n_k} - \frac{\vec{a} \cdot \vec{n}_k}{2}) (v_k) = g_{k-1}$ on $\Gamma_{k-1,k}$ for $2 \leq k \leq N$,
- (8) $v_k = 0$ on $\partial\Omega \cap \partial\Omega_k$.

REMARK 2. Our approach is different from that used in [7] or [4], where the interface conditions $\nu \frac{\partial}{\partial n_k} - \min(\vec{a} \cdot \vec{n}_k, 0)$ are used in the framework of Schwarz algorithms.

The Robin boundary conditions in (6)-(7) are not standard and lead nevertheless to a well-posed problem:

PROPOSITION 3. *Let ω be an open set of \mathbb{R}^2 , $f \in L^2(\omega)$, $\lambda \in H^{-1/2}(\partial\omega)$, $\vec{a} \in (C^1(\bar{\omega}))^2$, $c \in \mathbb{R}$ s.t. $c - \frac{1}{2} \operatorname{div}(\vec{a}) \geq \alpha > 0$ for some $\alpha \in \mathbb{R}$. Then, there exists a unique $u \in H^1(\omega)$ s.t.*

$$\begin{aligned} &\int \int_{\omega} c v w + (\vec{a} \cdot \nabla v) w + \nu \nabla v \cdot \nabla w - \int_{\partial\omega} \frac{\vec{a} \cdot \vec{n}}{2} v w \\ &= \langle \lambda, w \rangle_{H^{-1/2} \times H^{1/2}} + \int \int_{\omega} f w, \quad \forall w \in H^1(\omega). \end{aligned}$$

PROOF. When using the Lax-Milgram theorem, the only thing which is not obvious is the coercivity of the bilinear form

$$(v, w) \mapsto \int \int_{\omega} c v w + (\vec{a} \cdot \nabla v) w + \nu \nabla v \cdot \nabla w - \int_{\partial \omega} \frac{\vec{a} \cdot \vec{n}}{2} v w$$

Integrating by parts leads to

$$\begin{aligned} \int \int c v^2 + (\vec{a} \nabla v) v + \nu |\nabla v|^2 - \int_{\partial \omega} \frac{\vec{a} \cdot \vec{n}}{2} v^2 &= \int \int (c - \frac{1}{2} \operatorname{div}(\vec{a})) v^2 + \nu |\nabla v|^2 \\ &\geq \min(\alpha, \nu) \|v\|_{H^1(\omega)}^2. \end{aligned}$$

□

PROPOSITION 4. *In the case where the plane \mathbb{R}^2 is decomposed into the left ($\Omega_1 =]-\infty, 0[\times \mathbb{R}$) and right ($\Omega_2 =]0, \infty[\times \mathbb{R}$) half-planes and where the velocity \vec{a} is uniform, we have that*

$$\mathcal{T} \circ \mathcal{S}(., 0) = Id.$$

PROOF. A point in \mathbb{R}^2 is denoted by (x, y) . The vector \vec{a} is denoted $\vec{a} = (a_x, a_y)$. The unit outward normal and tangential vectors to domain Ω_k are denoted by \vec{n}_k and $\vec{\tau}_k$ respectively. The proof is based on the Fourier transform in the y direction and the Fourier variable is denoted by ξ . The inverse Fourier transform is denoted by \mathcal{F}^{-1} . Let us compute $\mathcal{S}(u_0, 0)$ for $u_0 \in H^{1/2}(\mathbb{R})$. Let w_k be the solution to (1)-(4), with $f = 0$ and u_0 as a Dirichlet data. The Fourier transform of (1) w.r.t. y yields

$$(c + a_x \partial_x + a_y i \xi - \nu \partial_{xx} + \nu \xi^2)(\hat{w}_k(x, \xi)) = 0$$

where $i^2 = -1$. For a given ξ , this equation is an ordinary differential equation in x whose solutions have the form $\alpha_k(\xi) e^{\lambda_k(\xi)|x|} + \beta_k(\xi) e^{\tilde{\lambda}_k(\xi)|x|}$ where

$$\lambda_k(\xi) = \frac{-\vec{a} \cdot \vec{n}_k - \sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_1 \xi \nu + 4\xi^2 \nu^2}}{2\nu}$$

and

$$\tilde{\lambda}_k(\xi) = \frac{-\vec{a} \cdot \vec{n}_k + \sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_1 \xi \nu + 4\xi^2 \nu^2}}{2\nu}$$

The solutions w_k must be bounded at infinity so that $\beta_k = 0$. The Dirichlet boundary conditions at $x = 0$ give $\alpha_k(\xi) = \hat{u}_0(\xi)$. Finally, we have that $w_k = \mathcal{F}^{-1}(\hat{u}_0(\xi) e^{\lambda_k(\xi)|x|})$ satisfy (1)-(3). Hence,

$$\mathcal{S}(u_0, 0) = \frac{1}{2} \mathcal{F}^{-1}(\sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_2 \xi \nu + 4\xi^2 \nu^2} \hat{u}_0(\xi)).$$

In the same way, it is possible to compute $\mathcal{T}(g)$ for $g \in H^{-1/2}(\mathbb{R})$. Indeed, let v_1 (resp. v_2) be the solution to (5)-(7) in domain Ω_1 (resp. Ω_2). The function v_k may be sought in the form $v_k = \mathcal{F}^{-1}(\alpha_k(\xi) e^{\lambda_k(\xi)|x|})$. The boundary conditions (6)-(7) give:

$$\begin{aligned} \hat{g}(\xi) &= (-\nu \lambda_k(\xi) - \frac{\vec{a} \cdot \vec{n}_k}{2}) \alpha_k(\xi) \\ &= \frac{\sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_2 \xi \nu + 4\xi^2 \nu^2}}{2} \alpha_k(\xi) \end{aligned}$$

Hence, $\hat{v}_k(0, \xi) = \frac{2}{\sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_2 \xi \nu + 4\xi^2 \nu^2}} \hat{g}(\xi)$ and $\widehat{\mathcal{T}(g)} = \frac{1}{2}(\hat{v}_1(0, \xi) + \hat{v}_2(0, \xi))$
i.e.

$$\mathcal{T}(g) = 2\mathcal{F}^{-1}\left(\frac{\hat{g}(\xi)}{\sqrt{4\nu c + (\vec{a} \cdot \vec{n}_k)^2 + 4i\vec{a} \cdot \vec{\tau}_2 \xi \nu + 4\xi^2 \nu^2}}\right).$$

Hence, it is clear that $\mathcal{T} \circ \mathcal{S}(., 0) = Id$. \square

REMARK 5. The same kind of computation shows that if $\max(\frac{cL}{|\vec{a} \cdot \vec{n}|}, L\sqrt{\frac{c}{\nu}}) \gg 1$, we still have $\mathcal{T} \circ \mathcal{S}(., 0) \simeq Id$. This means the preconditioner \mathcal{T} remains efficient for an arbitrary number of subdomains as long as the advective term is not too strong or the viscosity is small enough. Moreover, in the case of simple flows, we expect that the preconditioned operator is close to a nilpotent operator whose nilpotency is the number of subdomains, [2]. In this case, the convergence does not depend on the parameter c and the method works well for large δt .

3. The Discrete Case

We suppose for simplicity that the computational domain is \mathbb{R}^2 discretized by a Cartesian grid. Let us denote $A = (A_{ij}^{kl})_{i,j,k,l \in \mathbb{Z}}$ the matrix resulting from a discretization of the advection-diffusion problem. We suppose that the stencil is a 9-point stencil ($A_{ij}^{kl} = 0$ for $|i - k| \geq 2$ or $|j - l| \geq 2$). This is the case, for instance, for a Q1-SUPG method or for a classical finite difference or finite volume scheme. We have to solve $AU = F$ where $U = (u_{ij})_{i,j \in \mathbb{Z}}$ is the vector of the unknowns. The computational domain is decomposed into two half planes ω_1 and ω_2 . We introduce a discretized form \mathcal{S}_h of the operator \mathcal{S} (we adopt the summation convention of Einstein over all repeated indices)

$$(9) \quad \mathcal{S}_h : \mathbb{R}^{\mathbb{Z}} \times \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$$

$$(10) \quad ((u_{0j})_{j \in \mathbb{Z}}, (F_{ij})_{i,j \in \mathbb{Z}}) \mapsto (A_{0j}^{-1l} v_{-1l}^1 + B_j^1{}^l v_{0l}^1 + A_{0j}^{1l} v_{1l}^2 + B_j^2{}^l v_{0l}^2 - F_{0j})_{j \in \mathbb{Z}}$$

where (v_{ij}^m) satisfy

$$\begin{aligned} A_{ij}^{kl} v_{kl}^m &= F_{ij} && \text{for } i < 0 \text{ if } m = 1 \text{ and } i > 0 \text{ if } m = 2, \quad j \in \mathbb{Z}, \\ v_{0j}^m &= u_{0j}, && \text{for } j \in \mathbb{Z}. \end{aligned}$$

The coefficients $B_j^m{}^l$ are the contributions of the domain ω_m to A_{0j}^{0l} , $A_{0j}^{0l} = B_j^1{}^l + B_j^2{}^l$. For example, if $\vec{a} = 0$, $B_j^1{}^l = B_j^2{}^l = A_{0j}^{0l}/2$. For example, for a 1D case with a uniform grid and an upwind finite difference scheme ($a > 0$) and $c = 0$,

$$A_0^{-1} = -\frac{\nu}{h^2} - \frac{a}{h}, \quad A_0^0 = \frac{2\nu}{h^2} + \frac{a}{h}, \quad A_0^1 = -\frac{\nu}{h^2}, \quad B^1 = \frac{\nu}{h^2} + \frac{a}{h} \quad \text{and} \quad B^2 = \frac{\nu}{h^2}.$$

$\mathcal{S}_h(u_0, F)$ is the residual of the equation on the interface. It is clear that $U_0 = (u_{0j})_{j \in \mathbb{Z}}$ satisfies

$$(11) \quad \mathcal{S}_h(U_0, 0) = -\mathcal{S}_h(0, F).$$

We propose for an approximate inverse of $\mathcal{S}_h(., 0)$, \mathcal{T}_h defined by

$$(12) \quad \mathcal{T}_h : \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$$

$$(13) \quad (g_j)_{j \in \mathbb{Z}} \mapsto \frac{1}{2}(v_{0j}^1 + v_{0j}^2)_{j \in \mathbb{Z}},$$

where $(v_{ij}^1)_{i \leq 0, j \in \mathbb{Z}}$ and $(v_{ij}^2)_{i \geq 0, j \in \mathbb{Z}}$ satisfy

$$(14) \quad A_{ij}^{kl} v_{kl}^1 = 0, \quad i < 0, j \in \mathbb{Z}, \quad A_{0j}^{-1l} v_{-1l}^1 + \frac{A_{0j}^{0l}}{2} v_{0l}^1 = g_j$$

and

$$(15) \quad A_{ij}^{kl} v_{kl}^2 = 0, \quad i > 0, j \in \mathbb{Z}, \quad A_{0j}^{1l} v_{1l}^2 + \frac{A_{0j}^{0l}}{2} v_{0l}^2 = g_j.$$

REMARK 6. For a Neumann-Neumann preconditioner, in place of (14) and (15), we would have $A_{0j}^{-1l} v_{-1l}^1 + B_j^{-1l} v_{0l}^1 = g_j$ and $A_{0j}^{1l} v_{1l}^2 + B_j^{1l} v_{0l}^2 = g_j$.

REMARK 7. For a constant coefficient operator \mathcal{L} and a uniform grid, a discrete Fourier analysis can be performed similarly to that of the previous section. It can then be proved that

$$\mathcal{T}_h \circ \mathcal{S}_h(., 0) = Id_h$$

REMARK 8. The last equations of (14) and (15) correspond to the discretization of the Robin boundary condition $\nu \frac{\partial}{\partial n} - \frac{\vec{a} \cdot \vec{n}}{2}$. Considering the previous 1D example, we have

$$h(A_0^{-1} v_{-1}^1 + \frac{A_0^0}{2} v_0^1) = (\nu + ah/2) \frac{v_0^1 - v_{-1}^1}{h} - \frac{a}{2} v_{-1}^1 = \nu \frac{v_0^1 - v_{-1}^1}{h} - \frac{a}{2} (2v_{-1}^1 - v_0^1),$$

and

$$(16) \quad h(A_j^{1l} v_1^2 + \frac{A_0^0}{2} v_0^2) = \nu \frac{v_0^2 - v_1^2}{h} + \frac{a}{2} v_0^2.$$

Another discretization of $\nu \frac{\partial}{\partial n} - \frac{\vec{a} \cdot \vec{n}}{2}$ would not give (16). This is the reason why the approximate inverse is directly defined at the algebraic level. The discretization of the Robin boundary condition is in some sense adaptive with respect to the discretization of the operator (SUPG, upwind finite difference scheme or finite volume scheme). In the previous 1D example, a straight forward discretization of the Robin boundary condition would give for domain 1

$$\nu \frac{v_0^1 - v_{-1}^1}{h} - \frac{a}{2} v_0^1.$$

When $ah \gg \nu$, which is usually the case, it is quite different from (14).

4. Adaption to the Mortar Method

The mortar method was first introduced by C. Bernardi, Y. Maday and T. Patera ([3]). It has been extended to advection-diffusion problems by Y. Achdou ([1]). It enables to take nonmatching grids at the interfaces of the subdomains without loss of accuracy compared to matching grids. In our case, the additional difficulty lies in the equations (14) and (15) which are no longer defined. Indeed, the coefficients A_{0j}^{0l} are defined only for matching grids where they correspond to coefficients of the matrix before the domain decomposition. Only the coefficients B_j^{ml} are available. Then, the trick is to take for A_{0j}^{0l} in (14) and (15), the matrix entries at the nearest interior points of the subdomains. Therefore, the equations (14) and (15) are replaced by

$$A_{ij}^{kl} v_{kl}^1 = 0, \quad i < 0, j \in \mathbb{Z}, \quad A_{0j}^{-1l} v_{-1l}^1 + \frac{A_{-1j}^{-1l}}{2} v_{0l}^1 = g_j$$

and

$$A_{ij}^{kl} v_{kl}^2 = 0, \quad i > 0, j \in \mathbb{Z}, \quad A_{0j}^{1l} v_{1l}^2 + \frac{A_{1j}^{1l}}{2} v_{0l}^2 = g_j.$$

TABLE 1. Number of iterations for different domain decompositions ($\vec{a} = \min(300y^2, 3)e_1$)

	Precond.	40-20-40	40-40-40	40-60-40	60-60-60	60-60-60(geo)	60-60-60-60
$\delta t = 1$ $\nu = 0.001$	R-R	11	12	13	13	12	11
	N-N	31	37	38	43	67	60
	-	21	33	37	>100	66	>100
$\delta t = 0.1$ $\nu = 0.01$	R-R	13	13	12	11	11	10
	N-N	22	23	25	26	31	33
	-	16	22	24	27	>100	19

TABLE 2. Number of iterations for different velocity fields, a three-domain decomposition and 40 points on each interface.

Precond.	normal	parallel	rotating	oblique
R-R	10	2	11	11
N-N	25	2	13	27
-	21	>50	45	12

5. Numerical Results

The advection-diffusion is discretized on a Cartesian grid by a Q1-streamline-diffusion method ([6]). Nonmatching grids at the interfaces are handled by the mortar method ([1]). The interface problem (11) is solved by a preconditioned GMRES algorithm. The preconditioners are either of the type Robin-Robin (R-R), Neumann-Neumann (N-N) or the identity (-). In the test presented below, all the subdomains are squares of side 0.5. The figures in Tables 1 and 2 are the number of iterations for reducing the initial residual by a factor 10^{-10} .

In Table 1, the first five columns correspond to a three-domain decomposition and the last one to a five-domains partition. The grid in each subdomain is a $N \times N$ Cartesian grid, not necessarily uniform. The first line indicates the parameters N . For instance 40 – 20 – 40 means that the first and third subdomain have a 40×40 grid whereas the second subdomain has a 20×20 grid. In this case, the grids do not match at the interfaces. The grids are uniform except for the last but one column: in this case, the grid is geometrically refined in the y -direction with a ratio of 1.2. The velocity which is not varied, has a boundary layer in the y -direction.

In Table 2, the velocity field has been varied:

normal (to the interfaces): $\vec{a} = \min(300 * y^2, 3)e_1$, parallel (to the interfaces): $\vec{a} = e_2$, rotating: $\vec{a} = (y - y_0)e_1 - (x - x_0)e_2$ where (x_0, y_0) is the center of the computational domain and oblique: $\vec{a} = 3e_1 + e_2$. The mesh is fixed, the viscosity is $\nu = 0.01$ and the time step is $\delta t = \frac{1}{c} = 1$.

Tables 1 and 2 show that the proposed Robin-Robin preconditioner is very stable with respect to the mesh refinement, the number of subdomains, the aspect ratio of the meshes and the velocity field. More complete tests as well as the complete description of the solver will be given in [2].

6. Conclusion

We have proposed a preconditioner for the non symmetric advection-diffusion equation which generalizes the Neumann-Neumann preconditioner [5] in the sense that:

- It is exact for a two-domain decomposition.
- In the symmetric case, it reduces to the Neumann-Neumann preconditioner.

The tests have been performed on a decomposition into strips with various velocities and time steps. The results prove promising. In a forthcoming paper [2], we shall consider more general decompositions and the addition of a coarse level preconditioner.

References

1. Yves Achdou, *The mortar method for convection diffusion problems*, C.R. Acad. Sciences Paris, serie I **321** (1995), 117–123.
2. Yves Achdou, Patrick Lelallec, and Frédéric Nataf, *The Robin-Robin preconditioner for transport equations*, In preparation.
3. C. Bernardi, Y. Maday, and A. T. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991), Pitman Res. Notes Math. Ser., vol. 299, Longman Sci. Tech., Harlow, 1994, pp. 13–51.
4. F. Gastaldi, L. Gastaldi, and A. Quarteroni, *Adaptative domain decomposition methods for advection dominated equations*, East-West J. Numer. Math. **4** (1996), 165–206.
5. J.F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu, *Variational formulation and algorithm for trace operator in domain decomposition calculations*, Proceedings of DDM 2, AMS, 1988, pp. 3–16.
6. C. Johnson, U. Navert, and J. Pitkaranta, *Finite element method for linear hyperbolic problems*, Comp. Meth. in Appl. Eng. **45** (1984), 285–312.
7. Frédéric Nataf and François Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Math. Models Appl. Sci. **5** (1995), 67–93.

INSA RENNES, 20 AV DES BUTTES DE COESMES, 35043 RENNES, FRANCE
E-mail address: yves.achdou@insa-rennes.fr

CMAP UMR7641 CNRS, ECOLE POLYTECHNIQUE, 91128, PALAISEAU CEDEX, FRANCE
E-mail address: nataf@cmapx.polytechnique.fr

A Semi-dual Mode Synthesis Method for Plate Bending Vibrations

Frédéric Bourquin and Rabah Namar

1. Introduction

Once a structure is decomposed into substructures, mode synthesis is the Rayleigh-Ritz approximation of the global eigenvalue problem on the space spanned by a few eigenmodes of each substructure and some *coupling modes* which aim at describing the restriction to the interface of the global eigenmodes [7]. These *coupling modes* are defined here at the continuous level as the eigenfunctions of an *ad hoc* preconditioner of the Poincaré-Steklov operator associated with the interface as in [2]. The definition of this preconditioner of Neumann-Neumann type relies on suitable extension operators from the boundary of the subdomains to the whole interface. This paper concentrates on the definition of such extension operators in the case of plate bending and for general domain decompositions with cross-points and extends [2].

The plate bending problem is posed over a domain $\omega \subset \mathbb{R}^2$ which is splitted in p substructures ω_i separated by an interface γ . Let D , ν , and ρ denote the non-necessarily constant stiffness, Poisson's ratio and mass density respectively. Greek indices take their value in $\{1, 2\}$ and summation of repeated indices is assumed. For $u, v \in H^2(\omega)$, define :

$$(1) \quad \begin{cases} (u, v)_i = \int_{\omega_i} \rho u v, \\ a_i(u, v) = \int_{\omega_i} D((1 - \nu) \partial_{\alpha\beta} u \partial_{\alpha\beta} v + \nu \Delta u \Delta v) + d(u, v)_i, \\ (u, v) = \sum_{i=1}^p (u, v)_i, \quad a(u, v) = \sum_{i=1}^p a_i(u, v), \end{cases}$$

where d stands for an arbitrary positive constant and Δ for the Laplacian. Let V denote the space of admissible displacements, *i.e.* the subspace of $H^2(\omega)$ satisfying the Dirichlet boundary conditions along $\partial\omega$ of the problem, if any.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N25, 73K10.

The authors express their warmest thanks to J-J. Brioist and the whole team in charge of developing CESAR-LCPC for their help during the implementation of this method.

It is well-known that the global eigenvalue problem :

$$\text{Find } (\lambda, u) \in \mathbb{R} \times V \text{ s.t. } a(u, v) = \lambda(u, v) \quad \forall v \in V$$

possesses a family $(\lambda_k, u_k)_{k=1}^{+\infty}$ of solutions. In the same way, for $1 \leq i \leq p$ set :

$$\gamma_i = \gamma \cap \partial\omega_i, \quad V_i = \{w|_{\omega_i}; w \in V\} \quad \text{and} \quad V_i^0 = \{v \in V_i; v = 0 \text{ and } \nabla v = 0 \text{ on } \gamma_i\}.$$

Let $(\lambda_{ij}, u_{ij})_{j=1}^{+\infty} \in \mathbb{R}^+ \times V_i^0$ denote the family of solutions of the problem :

$$\begin{aligned} \text{Find } (\lambda, u) \in (R) \times V_i^0 &\text{ s.t.} \\ a_i(u, v) &= \lambda(u, v)_i \quad \forall v \in V_i^0. \end{aligned}$$

Mode synthesis uses as test functions the *fixed interface modes* u_{ij} and *coupling modes* that do not identically vanish on γ .

2. Definition of the coupling modes

2.1. Basic trace and extension properties. The admissible displacements $w \in H^2(\omega)$ possess two independent traces along γ , $w|_\gamma$ and $\theta_n = (\frac{\partial w}{\partial \vec{n}})|_\gamma$, where \vec{n} denotes a unit normal vector along γ , that is defined on each edge independently. Let us recall a characterization of the space $V_\gamma = \left\{ w|_\gamma, (\frac{\partial w}{\partial \vec{n}})|_\gamma; w \in V \right\}$: along each edge Γ_i , $(w|_\gamma, (\frac{\partial w}{\partial \vec{n}})|_\gamma) \in H^{3/2}(\Gamma_i) \times H^{1/2}(\Gamma_i)$. Moreover, compatibility conditions hold at every vertex \mathcal{O} of the interface. Assume that two edges Γ_1 and Γ_2 share a common vertex \mathcal{O} , as in Figure 1, and denote by $(w_1, \theta_{n,1})$ (resp. $(w_2, \theta_{n,2})$) the traces of w along Γ_1 (resp. Γ_2). The continuity of w and the $H^{1/2}$ -continuity of ∇w must be ensured at \mathcal{O} . Since $\nabla w|_{\Gamma_i} = \frac{\partial w_i}{\partial s_i} \vec{\tau}_i + \theta_{n,i} \vec{n}_i$ if $\vec{\tau}_i$ denotes a unit tangential vector along Γ_i , s_i an associated curvilinear abscissa on Γ_i , and $\vec{n}_i = \vec{n}|_{\Gamma_i}$, the compatibility conditions write :

$$(2) \quad \begin{cases} w_1(\mathcal{O}) &= w_2(\mathcal{O}) \\ \frac{\partial w_1}{\partial s_1} \vec{\tau}_1 + \theta_{n,1} \vec{n}_1 &= \frac{\partial w_2}{\partial s_2} \vec{\tau}_2 + \theta_{n,2} \vec{n}_2 \quad \text{at } \mathcal{O} \end{cases}$$

for every vertex \mathcal{O} and every set of edges that cross at \mathcal{O} . If N_e denotes the number of edges, it turns out that V_γ is isomorphic to the subspace of $\prod_{i=1}^{N_e} H^{3/2}(\Gamma_i) \times H^{1/2}(\Gamma_i)$ of pairs satisfying above compatibility conditions as well as the Dirichlet boundary conditions along $\partial\omega$, if any.

Let $\mathcal{R} : V_\gamma \rightarrow V$ denote the *biharmonic* extension operator defined by

$$(3) \quad \begin{cases} a(\mathcal{R}(v, \theta), z) = 0 \quad \forall z \in V^0 = \bigcup_{i=1}^p V_i^0, \\ \mathcal{R}(v, \theta) = v, \quad \frac{\partial \mathcal{R}(v, \theta)}{\partial \vec{n}} = \theta \quad \text{along } \gamma. \end{cases}$$

This problem splits into p independent plate problems. Now, let $(w_l, \theta_{n,l})_{l=1}^{+\infty}$ denote a given dense family in V_γ . Then *coupling modes* will be defined as $\mathcal{R}(w_l, \theta_{n,l}) \in V$. The question of choosing a suitable family is addressed in the next section.

2.2. A generalized Neumann-Neumann preconditioner. Let us set $V_{\gamma_i} = \{(v, \theta)_{|\gamma_i}; (v, \theta) \in V_\gamma\}$, and let $P_i : V_{\gamma_i} \longrightarrow V_\gamma$ denote a continuous extension operator, that is also defined as a continuous operator from $L^2(\gamma_i) \times L^2(\gamma_i)$ to $L^2(\gamma) \times L^2(\gamma)$. For any pair of functions (T, M) defined on γ_i , let $w \in V_i$ stand for the solution of the well-posed Neumann problem

$$a_i(w, z) = \int_{\gamma_i} Tz + M \frac{\partial z}{\partial \vec{n}} \quad \forall z \in V_i,$$

and define the mapping S_i by $S_i(T, M) = (w|_{\gamma_i}, \frac{\partial w}{\partial \vec{n}}|_{\gamma_i})$. This local dual Schur complement $S_i : V'_{\gamma_i} \longrightarrow V_{\gamma_i}$ is continuous.

Then the operator $S : V'_\gamma \longrightarrow V_\gamma$, $S = \sum_{i=1}^p P_i S_i P_i^*$ is continuous. It is compact over $L^2(\gamma) \times L^2(\gamma)$ and symmetric, hence it possesses a family of finite-dimensional eigenspaces associated with positive decreasing eigenvalues $(\mu_{\gamma\ell})_{\ell=1}^{+\infty}$ and also a possibly infinite-dimensional kernel, $\ker(S)$. Since by construction the operators S_i are isomorphisms, $\ker(S) = \ker(\sum_{i=1}^p P_i P_i^*)$. Therefore, the family chosen of *coupling modes* naturally splits in two subfamilies :

- The first one is made of the *biharmonic* extensions $\mathcal{R}(w_\ell, \theta_\ell)$ of a given number N_γ of independent eigenfunctions (w_ℓ, θ_ℓ) associated with the largest eigenvalues $\mu_{\gamma\ell}$.

- As for the second one, noticing that

$$\ker\left(\sum_{i=1}^p P_i P_i^*\right) = \bigcap_{i=1}^p \ker(P_i P_i^*) \subset \bigcup_{i=1}^p \ker(P_i P_i^*)$$

we decide to retain the low frequency content of each subspace $\ker(P_i P_i^*)$, namely the M_i first solutions of the auxiliary eigenvalue problem

$$(4) \quad \int_{\gamma_i} \frac{\partial^2 w}{\partial \tau_i^2} \frac{\partial^2 v}{\partial \tau_i^2} + \int_{\gamma_i} \frac{\partial \theta}{\partial \tau_i} \frac{\partial \psi}{\partial \tau_i} + \frac{1}{\epsilon} \int_{\gamma_i} P_i^*(w, \theta) P_i^*(v, \psi) = \xi \int_{\gamma_i} wv + \theta\psi$$

$\forall (v, \psi) \in H^2(\gamma) \times H^1(\gamma)$, where ϵ is a small parameter.

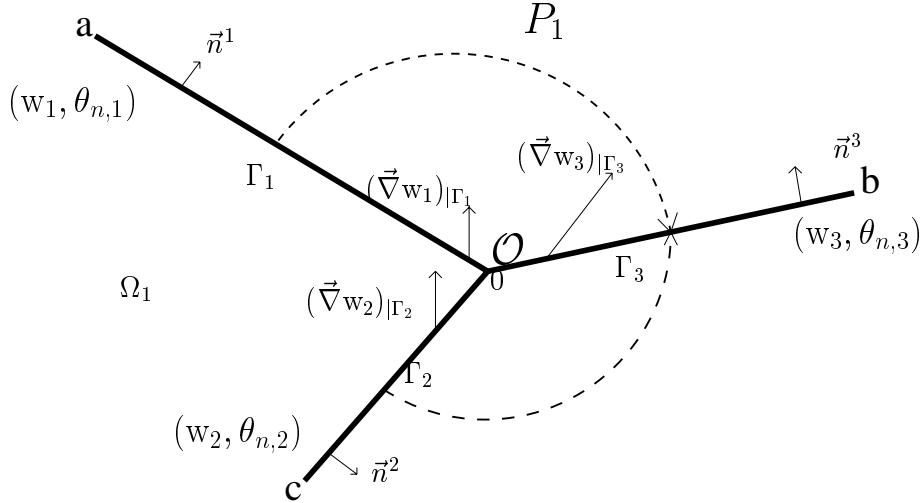
The *biharmonic* extensions $\mathcal{R}(\tilde{w}_{ij}, \tilde{\theta}_{ij})$ of these $M = \sum_{i=1}^p M_i$ modes $(\tilde{w}_{ij}, \tilde{\theta}_{ij})$ form the second family of *coupling modes*. The resulting mode synthesis method amounts to define the finite-dimensional space

$$V_N = \text{span} \left\{ \bigcup_{i=1}^p (u_{ij})_{j=1}^{N_i} \bigcup (\mathcal{R}(w_\ell, \theta_\ell))_{\ell=1}^{N_\gamma} \bigcup_{i=1}^p (\mathcal{R}(\tilde{w}_{ij}, \tilde{\theta}_{ij}))_{j=1}^{M_i} \right\},$$

for some numbers N_i , M_i , N_γ , $1 \leq i \leq p$, and to perform the Galerkin approximation of the global eigenvalue problem on this space.

REMARK 1. it is possible to further filter the kernels $\ker(P_i P_i^*)$ by just solving the eigenvalue problem (4) again after projection over the solutions of the first solve and with $\epsilon = +\infty$

REMARK 2. If the extension operator preserves some locality, this eigenvalue problem is posed over a limited set of edges, not on the whole interface. Moreover, since P_i only depends on the geometry of the interface and, at the discrete level, on the mesh, the auxiliary eigenvalue problem (4) does not require any subdomain solve, and, in practice, proves very cheap.

FIGURE 1. The extension operator by reflection for H^2 functions.

REMARK 3. Mode synthesis appears as a non iterative domain decomposition method contrary to [5], [6]. The proposed method differs from [1] where the Schur complement is used instead of a preconditioner. It also differs from [4] since the preconditioner is not used to compute the spectrum of the Schur complement.

The next section is devoted to the construction of the extension operators P_i .

3. The extension operators $\mathbf{P}_i : \mathbf{V}_{\gamma_i} \rightarrow \mathbf{V}_{\gamma}$

Let $w \in V_i$ denote some given function. Its traces $(w|_{\gamma_i}, \frac{\partial w}{\partial \vec{n}}|_{\gamma_i})$ are to be extended onto the adjacent edges of the interface. First pick out two edges Γ_1 and Γ_2 of γ_i that share a common vertex O , and choose an adjacent edge $\Gamma_3 \not\subset \gamma_i$, as in Figure 1. Parametrize Γ_1 (resp. Γ_2, Γ_3) by the curvilinear abscissa $s_1 \in [a, 0]$ (resp. $s_2 \in [c, 0], s_3 \in [0, b]$), starting from the vertex O . As in section 2, let $(w_1, \theta_{n,1})$ (resp. $(w_2, \theta_{n,2})$) stand for the traces of w along Γ_1 (resp. Γ_2). These traces satisfy the compatibility conditions (2). A pair of traces $(w_3, \theta_{n,3})$ is sought on Γ_3 in such a way that the compatibility conditions

$$(5) \quad \begin{cases} w_3(O) &= w_1(O) \\ \frac{\partial w_3}{\partial s_3} \vec{\tau}_3 + \theta_{n,3} \vec{n}_3 &= \frac{\partial w_1}{\partial s_1} \vec{\tau}_1 + \theta_{n,1} \vec{n}_1 \end{cases} \quad \text{at } O$$

hold for every possible value of $w_1(O)$, $\frac{\partial w_1}{\partial s_1}(O)$ and $\theta_{n,1}(O)$. The resulting traces will then coincide with the traces of a function in $H^2(\omega)$.

The compatibility conditions (5) lead to 5 scalar equations. This is why the proposed extension operator P_i involves 5 parameters to be identified :

$$\left(P_i \left(w|_{\gamma_i}, \frac{\partial w}{\partial \vec{n}}|_{\gamma_i} \right) \right)_{|\Gamma_3} = (w_3, \theta_{n,3}) \quad \text{with}$$

$$(6) \quad \begin{cases} w_3(s) &= \varphi(s) \left\{ \frac{\alpha_{13}}{2} w_1\left(\frac{a}{b}s\right) + \beta_{13} w_1\left(\frac{2a}{b}s\right) + \frac{\alpha_{13}}{2} w_2\left(\frac{c}{b}s\right) + \beta_{23} w_2\left(\frac{2c}{b}s\right) \right\} \\ \theta_{n,3}(s) &= \varphi(s) \left\{ \eta_{13}\theta_{n,1}\left(\frac{a}{b}s\right) + \eta_{23}\theta_{n,2}\left(\frac{c}{b}s\right) \right\}, \end{cases}$$

and where φ denotes some cut-off function whose support forms a neighborhood of \mathcal{O} .

Expressing the compatibility conditions leads to the linear system $A_3 X_3 = F_3$, where :

$$(7) \quad \begin{aligned} X_3 &= \begin{bmatrix} \alpha_{13} \\ \beta_{13} \\ \beta_{23} \\ \eta_{13} \\ \eta_{23} \end{bmatrix} \\ A_3 &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ \frac{a}{2b}\tau_{3,x} & \frac{2a}{b}\tau_{3,x} & 0 & C_1 n_{3,x} & n_{3,x}[C_1 \frac{n_{1,x}}{n_{2,x}} + \frac{\tau_{1,x}}{n_{2,x}}] \\ \frac{c}{2b}\tau_{3,x} & 0 & \frac{2c}{b}\tau_{3,x} & C_2 n_{3,x} & n_{3,x}[C_2 \frac{n_{1,x}}{n_{2,x}} - \frac{\tau_{2,x}}{n_{2,x}}] \\ \frac{a}{2b}\tau_{3,y} & \frac{2a}{b}\tau_{3,y} & 0 & C_1 n_{3,y} & n_{3,y}[C_1 \frac{n_{1,x}}{n_{2,x}} + \frac{\tau_{1,x}}{n_{2,x}}] \\ \frac{c}{2b}\tau_{3,y} & 0 & \frac{2c}{b}\tau_{3,y} & C_2 n_{3,y} & n_{3,y}[C_2 \frac{n_{1,x}}{n_{2,x}} - \frac{\tau_{2,x}}{n_{2,x}}] \end{bmatrix} \\ F_3 &= \begin{bmatrix} 1 \\ \tau_{1,x} + C_1 n_{1,x} \\ C_2 n_{1,x} \\ \tau_{1,y} + C_1 n_{1,y} \\ C_2 n_{1,y} \end{bmatrix} \end{aligned}$$

with

$$C_1 = \frac{\tau_{1,x} n_{2,y} - \tau_{1,y} n_{2,x}}{n_{1,y} n_{2,x} - n_{1,x} n_{2,y}}, \quad C_2 = \frac{\tau_{2,y} n_{2,x} - \tau_{2,x} n_{2,y}}{n_{1,y} n_{2,x} - n_{1,x} n_{2,y}}$$

and

$$\tau_{\ell,x} = \vec{\tau}_\ell \cdot \vec{x}, \quad \tau_{\ell,y} = \vec{\tau}_\ell \cdot \vec{y}, \quad n_{\ell,x} = \vec{n}_\ell \cdot \vec{x}, \quad n_{\ell,y} = \vec{n}_\ell \cdot \vec{y}, \quad 1 \leq \ell \leq 3.$$

From symbolic calculus the determinant is equal to $2\frac{ac}{b}$. This system is solved once and for all, thus yielding the parameters of the extension operator.

This process is repeated for all adjacent edges containing \mathcal{O} , and for all vertices. It follows from (6) that compatibility will also hold among all edges onto which the original traces are extended. Therefore $P_i : V_{\gamma_i} \rightarrow V_\gamma$ is continuous. Moreover, it is clear from its definition that $P_i : L^2(\gamma_i) \times L^2(\gamma_i) \rightarrow L^2(\gamma) \times L^2(\gamma)$ is also continuous see [3] for details.

4. Numerical tests

A square plate is decomposed into 9 subdomains and discretized with 15000 dof. Since we focus on the coupling strategy, a large number of *fixed interface modes* is used. Notice that only 17 modes of the dual Schur complement and less than 3 modes of each kernel $\ker(P_i P_i^*)$ are sufficient to yield a 1% accuracy on the first 20 eigenfrequencies (Fig. 2).

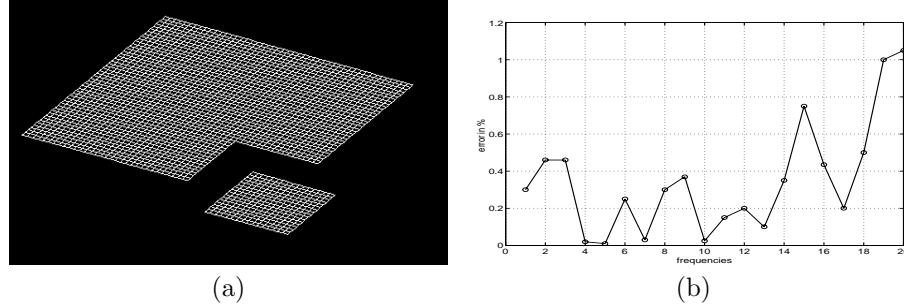


FIGURE 2. (a): The mesh and a typical subdomain, (b): the accuracy for $N_\gamma = 17$ and $M = \sum_{i=1}^p M_i = 23$

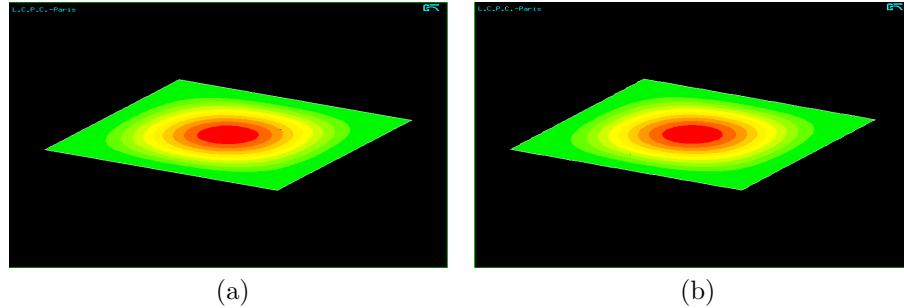


FIGURE 3. The first mode shape computed with : (a) A global F.E.M., and (b) Mode Synthesis for $N_\gamma = 17$ and $M = 23$

The mode shapes are also very accurate and smoothness is achieved, mainly because the extension operators are defined at the continuous level (Fig. 3, 4, and 5).

5. Concluding remarks

A new mode synthesis method is proposed. It can be formulated at the continuous level and at the discrete level. It is based on a *generalized* Neumann-Neumann preconditioner. It yields accurate frequencies and smooth mode shapes with a small number of *coupling modes* even when the interface exhibits cross-points. Its numerical analysis remains fairly open. See [3] for details.

References

1. F. Bourquin and F. d'Hennezel, *Intrinsic component mode synthesis and plate vibrations*, Comp. and Str. **44** (1992), no. 1, 315–324.
2. F. Bourquin and R. Namar, *Decoupling and modal synthesis of vibrating continuous systems*, Proc. Ninth Int. Conf. on Domain Decomposition Meths., 1996.
3. ———, *Extended Neumann-Neumann preconditioners in view of component mode synthesis for plates*, (1998), in preparation.
4. J. Bramble, V. Knyazev, and J. Pasciak, *A subspace preconditioning algorithm for eigenvalue/eigenvector computation*, Tech. report, University of Colorado at Denver, Center for Computational Mathematics, 1995.

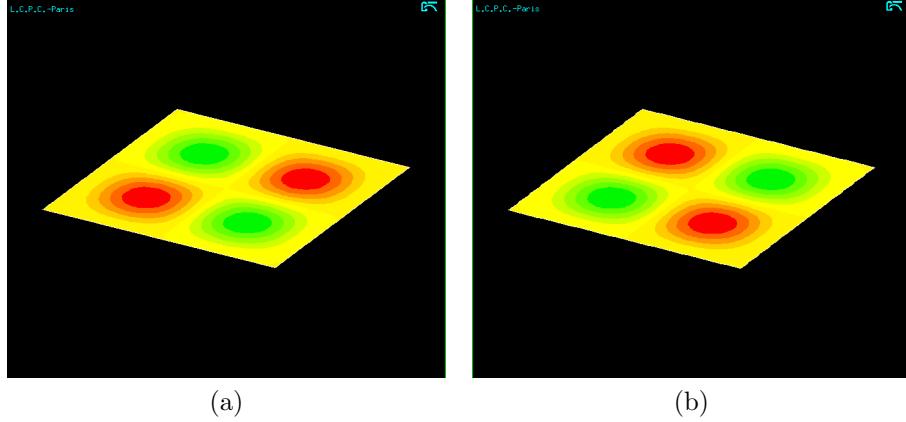


FIGURE 4. The 20th mode shape computed with : (a) A global F.E.M., and (b) Mode Synthesis for $N_\gamma = 17$ and $M = 23$

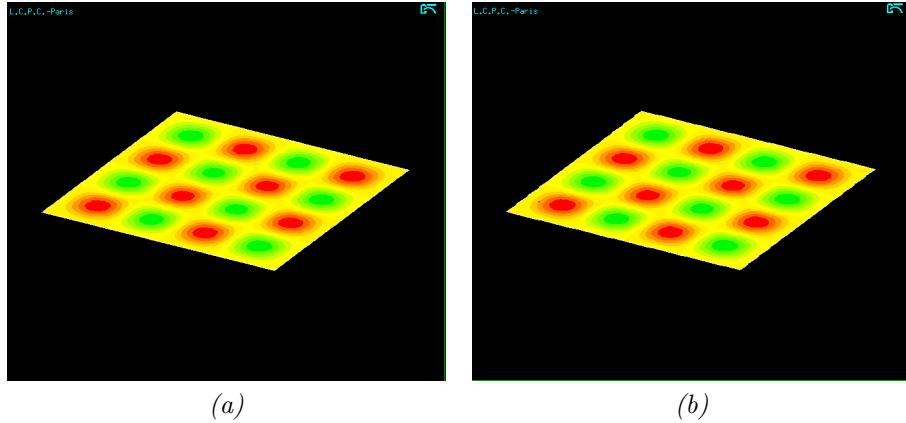


FIGURE 5. The 20th mode shape computed with : (a) A global F.E.M., and (b) Mode Synthesis for $N_\gamma = 17$ and $M = 23$

5. S. Maliassov, *On the analog of the Schwarz method for spectral problems*, Numerical Methods and Mathematical Modeling, Inst. Numer. Math., Russian Acad. Sci., Moscow (1992), 71–79, in Russian.
6. A. Sharapov and T. Chan, *Domain decomposition and multilevel methods for eigenvalue problems*, Proc. Ninth Int. Conf. on Domain Decomposition Meths., 1996.
7. D-M. Tran, *Méthodes de synthèse modale mixtes*, Revue Européenne des Eléments Finis **1** (1992), no. 2, 137–179.

LABORATOIRE DES MATÉRIAUX ET DES STRUCTURES DU GÉNIE CIVIL, UMR 113 LCPC-CNRS, 2 ALLÉE KEPLER, 77420 CHAMPS SUR MARNE, FRANCE

LABORATOIRE DES MATÉRIAUX ET DES STRUCTURES DU GÉNIE CIVIL, UMR 113 LCPC-CNRS, 2 ALLÉE KEPLER, 77420 CHAMPS SUR MARNE, FRANCE

Overlapping Schwarz Algorithms for Solving Helmholtz's Equation

Xiao-Chuan Cai, Mario A. Casarin, Frank W. Elliott, Jr.,
and Olof B. Widlund

1. Introduction

In this paper, we give a progress report on the development of a new family of domain decomposition methods for the solution of Helmholtz's equation. We present three algorithms based on overlapping Schwarz methods; in our favorite method we proceed to the continuous finite element approximation of the Helmholtz's equation through a sequence of discontinuous iterates. While this is, quite possibly, a new type of overlapping Schwarz methods, we have been inspired to develop this idea by the thesis of Després [4].

The basic domain decomposition algorithm considered by Després is defined as follows: The given region Ω is divided into two nonoverlapping subregions Ω_1 and Ω_2 , and the iteration is advanced by simultaneously solving

$$\begin{aligned} -\Delta u_j^{n+1} - k^2 u_j^{n+1} &= f \quad x \in \Omega_j, \\ (1) \quad \partial u_j^{n+1} / \partial n_{int} - iku_j^{n+1} &= -\partial u_{out}^n / \partial n_{out} - iku_{out}^n \quad x \in \Gamma, \\ \partial u_j^{n+1} / \partial n_{int} - iku_j^{n+1} &= g \quad x \in \partial\Omega, \end{aligned}$$

in the two subregions. Here f and g are data given for the original problem, k is a real parameter, and Γ the interface, i.e. the parts common to $\partial\Omega_1$ and $\partial\Omega_2$, the boundaries of subregions. We note that Sommerfeld-type boundary conditions are used and that the subregions themselves can be the union of a number of disjoint regions as in the case when Ω is cut into strips and the strips colored using two

1991 *Mathematics Subject Classification*. Primary 41A10; Secondary 65N30, 65N35, 65N55.

The first author was supported in part by the National Science Foundation under Grants ASC-9457534, ECS-9527169, and ECS-9725004, and in part by AFOSR Grant F49620-97-1-0059.

The second author was supported in part by the National Science Foundation under Grant NSF-ECS-9527169, and in part by CNPq Brazil.

The third author was supported in part by the National Science Foundation under Grant NSF-ECS-9527169, and in part by the U.S. Department of Energy under Contract DE-FG02-92ER25127.

The fourth author was supported in part by the National Science Foundation under Grants NSF-CCR-9503408 and NSF-ECS-9527169, and in part by the U.S. Department of Energy under Contract DE-FG02-92ER25127.

colors. The iterates generally have jumps across the interface; the jump will go to zero as the iteration converges.

In his thesis, Després proves convergence in a relatively weak sense for a quite general decomposition into nonoverlapping subregions and also conducts a detailed theoretical and numerical study for a rectangular Ω cut into two. The convergence is slow, but it is shown that under-relaxation can lead to an improvement. Després also briefly considers the use of overlap; it is shown that this leads to a considerable improvement in the rate of convergence of the iteration for the two subregion case.

In the limit of increasing domain diameter, the Sommerfeld boundary condition provides the correct far-field condition for propagation of waves in the frequency domain, but for a bounded region it does not provide perfect transparency. It can be argued that an alternative boundary condition, which more closely approximates the correct nonlocal non-reflecting boundary condition, would lead to more rapid convergence. These ideas have indeed been tested with some success by Ghanemi [5] and others. This essentially amounts to replacing the ik terms in the interface condition (1) with ikT , where T is an appropriate nonlocal operator. See also [9] for work more closely related to ours.

In our own work, we have instead attempted to use three ideas that have proven successful in studies of other types of problems: We have focused almost exclusively on methods based on overlapping decompositions of the region Ω . In addition, we are exploring the possible benefits of a coarse solver as a part of our preconditioner; we note that the use of a coarse space correction is required to establish convergence of domain decomposition algorithms for a class of nonsymmetric and indefinite elliptic problems previously considered by Cai and Widlund [2, 3]. We also take advantage of well-known accelerators of the basic iteration schemes, in particular the GMRES algorithm.

We refer to Smith, Bjørstad, and Gropp [11] for an introduction to domain decomposition methods in general, and these ideas in particular. We note that there are a number of variants of the Schwarz algorithms: additive, hybrid, restricted, etc. In our work, we are now focusing on the classical, multiplicative algorithm.

2. Differential and Discrete Model Problems

We consider a Helmholtz model problem given by

$$(2) \quad -\Delta u - k^2 u = f \quad x \in \Omega, \quad \partial u / \partial n - iku = g \quad x \in \partial\Omega,$$

where Ω is a bounded two or three-dimensional region. This equation is uniquely solvable, and we note that the boundary condition, said to be of Sommerfeld type, is essential in the proof of this fact.

We use Green's formula, and complex conjugation of the test functions, to convert (2) into variational form: Find $u \in H^1(\Omega)$ such that,

$$\begin{aligned} b(u, v) &= \int_{\Omega} (\nabla u \cdot \nabla \bar{v} - k^2 u \bar{v}) dx - ik \int_{\partial\Omega} u \bar{v} ds \\ &= \int_{\Omega} f \bar{v} dx + \int_{\partial\Omega} g \bar{v} ds = F(v) \quad \forall v \in H^1(\Omega). \end{aligned}$$

Finite element problems can now be defined straightforwardly by replacing $H^1(\Omega)$ by a suitable conforming finite element space. So far, we have worked mainly with

lower order elements but have made progress towards extending our studies and numerical experiments to spectral elements.

Our interest in the spectral element case has been inspired by the work of Ihlenburg and Babuška [7, 8, 6] and the thesis by Melenk [10]. They have considered the well-posedness of the original problem and different finite element discretizations and proven, for a model problem in one dimension, that the basic estimate

$$|u|_{H^1} \leq Ck|F|_{H^{-1}}$$

holds. In the finite element case, an assumption of $hk < 1$ is used. The constant C is independent of p , the degree of the finite elements. Ihlenburg has also conducted extensive numerical experiments which suggest that this bound also holds for problems in two or three dimensions. Error bounds of the following form are also given for $p = 1$ and kh small enough:

$$|\text{error}|_{H^1} \leq C_1\theta + C_2k\theta^2 \text{ where } \theta = \text{best } H^1\text{-error.}$$

With oscillatory solutions typical, we can expect θ to be on the order of kh . In that case, the second term, which is due to the phase error, will dominate unless k^2h is on the order of 1. Larger values of p appear attractive since Ihlenburg and Babuška have also shown that

$$|\text{error}|_{H^1} \leq \theta^p(C_1 + C_2\theta^2) + C_3k\theta^{2p}.$$

Here $\theta = hk/2p$, and the phase error is now relatively less important.

3. Overlapping Schwarz Algorithms

The basic multiplicative, one-level overlapping Schwarz method can be described as follows: Let $\{\Omega_j\}$ be a set of open subregions that covers the given region Ω . Just as in the strip case of Section 1, each subregion Ω_j can have many disconnected components; it is often profitable to color the subregions of an original overlapping decomposition of Ω using different colors for any pair of subregions that intersect. The original set of subregions can then be partitioned into sets of subregions, one for each color, effectively reducing the number of subregions. This decreases the number of fractional steps of our Schwarz methods and helps make the algorithms parallel. The number of colors is denoted by J .

In many cases, it is appropriate to view a multiplicative Schwarz method as follows: A full iteration step proceeds through J fractional steps,

$$u^{n+j/J} - u^{n+(j-1)/J} = P_j(u - u^{n+(j-1)/J}),$$

where $P_j, j = 1, \dots, J$, is a projection onto a subspace V_j related to Ω_j and u is the exact finite element solution. Such a fractional step can be more easily understood by rewriting it in the form

$$(3) \quad b_j(u^{n+j/J} - u^{n+(j-1)/J}, v) = F(v) - b(u^{n+(j-1)/J}, v) \quad \forall v \in V_j.$$

The choice of the local sesquilinear form $b_j(\cdot, \cdot)$ and the space V_j determines the projection P_j . We will examine several choices one of which has discontinuous iterates, and for it we will need an alternative to formula (3). Introducing a splitting of the Helmholtz form with respect to each Ω_j and its complement Ω_j^c ,

$$(4) \quad b(u, v) = b_j(u, v) + b_j^c(u, v),$$

which we will further describe below, we can replace (3) by

$$(5) \quad b_j(u^{n+j/J}, v) = F(v) - b_j^c(u^{n+(j-1)/J}, v).$$

It is also easy to introduce a coarse space correction and a second level into the algorithm. An additional fractional step is then used; we choose to make this correction prior to the other, local steps. In our experiments, we have so far only used the same low order finite element method on a coarser mesh. The space related to this mesh and fractional step is denoted by V_0 , and we use formula (3) to define the related, special update. We note that all the solvers used in the fractional steps are smaller, often much smaller, instances of the original problem.

One difficulty with faithfully implementing a generalization of Després' algorithm with overlap is the appearance and disappearance of multiple values, i.e. jumps, across different parts of the interface Γ , which is now defined by

$$\cup \partial\Omega_i \setminus \partial\Omega.$$

In our first two algorithms, we avoid jumps and use traditional domain decomposition techniques, but in the third and most successful algorithm jumps in the solution are fully accommodated.

The three algorithms can now be defined in terms of the sesquilinear forms $b_j(\cdot, \cdot)$ and the subspaces V_j .

ALG1 An update with zero Dirichlet condition on $\partial\Omega_j \setminus \partial\Omega$ is used in the j th fractional step; this preserves the continuity of the iterates. The test functions of V_j then vanish at all mesh points in the closure of Ω_j^c . The sesquilinear form is defined by

$$b_j(u, v) = \int_{\Omega_j} (\nabla u \cdot \nabla \bar{v} - k^2 u \bar{v}) dx - ik \int_{\partial\Omega \cap \partial\Omega_j} u \bar{v} ds.$$

We note that for an interior subregion we cannot guarantee solvability of the subproblem except by making the diameters of the components of the subregion Ω_j small enough. The same preconditioner can also be obtained by a matrix splitting based on the diagonal blocks of variables associated with the nodes in Ω_j and on $\partial\Omega \cap \partial\Omega_j$.

ALG2 The sesquilinear form is chosen as

$$b_j(u, v) = \int_{\Omega_j} (\nabla u \cdot \nabla \bar{v} - k^2 u \bar{v}) dx - ik \int_{\partial\Omega_j} u \bar{v} ds,$$

and the elements of the space V_j are now required to vanish at all the nodes in the open set Ω_j^c . We require that the solution of equation (5) belong to the same space. Continuity of the iterates is maintained by overwriting all the old values at all the nodes of the closure of Ω_j .

ALG3 The same V_j and $b_j(\cdot, \cdot)$ are used as in ALG2, but both the old and the new values on $\partial\Omega_j$ are saved. This will typically produce a jump across this part of the interface. At the same time the jump across the interface interior to Ω_j is eliminated. Further details will be given; we note that the new features of this algorithm have required a redesign of our data structures, and that there are consequences of the jumps that need careful scrutiny in order to understand ALG3 correctly.

Since we use completely standard techniques for the coarse grid correction, we describe only the fine grid fractional steps of ALG3 in some detail. We must first

realize that the lack of continuity across the interface Γ forces us to use broken norms, i.e. to replace integrals over Ω and the Ω_j by sums of integrals over *atomic* subregions defined by Γ . We proceed by finding the common refinement of all splittings like (4). Let $\{A_q | q = 1, \dots, Q\}$ be the *open atoms* generated by $\{\Omega_j\}$, i.e. the collection of the largest open sets satisfying $A_q \subseteq \Omega_j$ or $A_q \subseteq \Omega_j^c$ for all j and q . We refine (4) by expressing each term as a sum of Helmholtz forms defined on the collection of open atoms contained in that region. Thus,

$$\begin{aligned} b_j(u, v) &= \sum_{A_q \subseteq \Omega_j} b_q(u, v) \\ b_j^c(u, v) &= \sum_{A_q \subseteq \Omega_j^c} b_q(u, v). \end{aligned}$$

These are the splittings needed to solve equation (5) in the presence of jumps and to represent the solution in atomic form for further steps. The sesquilinear forms corresponding to the individual atoms are defined by

$$\underline{b}_q(u, v) = a_{A_q}(u, v) - ik(u, v)_{\tilde{\Xi}_q} - ik(u, v)_{\tilde{\Gamma}_q^-} + ik(u, v)_{\tilde{\Gamma}_q^+}$$

where,

$$\begin{aligned} \tilde{\Xi}_q &= \partial A_q \cap \partial \Omega \\ \tilde{\Gamma}_q^+ &= \bigcup \left\{ \partial A_q \cap \partial \Omega_j \mid \Omega_j^c \supseteq A_q, j = 1, \dots, J \right\} \\ \tilde{\Gamma}_q^- &= \bigcup \left\{ \partial A_q \cap \partial \Omega_j^c \mid \Omega_j \supseteq A_q, j = 1, \dots, J \right\}. \end{aligned}$$

For a valid splitting, we have to assume that the boundaries of any two intersecting subdomains, Ω_i and Ω_j , must have the same unit normal where they intersect, except on sets of measure zero.

The principal difficulty in implementing a multiplicative Schwarz cycle based on (5) is that it requires that multiple values be kept at the atom interfaces $\tilde{\Gamma}_q^-$ and $\tilde{\Gamma}_q^+$ because continuity is not enforced. Therefore, we represent the solution function u as an element in the direct product of finite element spaces, one for each atom. At iteration $n + j/J$ of the algorithm, see (5), the right hand side is computed atomic subregion by atomic subregion. It is therefore practical to store the nodal values of each atom separately. We also note that the test functions v are continuous functions in the closure of Ω and that the solution $u^{n+j/J}$ of (5) is continuous in the closure of Ω_j . Once it is found, it is scattered to the individual atoms of Ω_j . The set of nodal values of the iterate is exactly what is required in the computation of the contribution from that atom to the next set of right hand sides.

After a full sweep through all the subregions, the residuals interior to each atomic subregion are zero, and across any segment of Γ , the solution is either continuous or satisfies the flux condition. Then, by Green's formula, the approximate solution u^n satisfies

$$b(u^n, v) = F(v) + ik \int_{\Gamma} [u^n] \bar{v} ds \quad \forall v \in V.$$

In the limit, the jump goes to zero; this conveniently signals convergence. At this point of the iteration, the residuals can be computed from the jump directly, but also conventionally through its contributions from the atomic subregions.

4. Theoretical Results

Optimal convergence has been established for ALG1 and ALG2, with a coarse space V^H and GMRES acceleration, using essentially only our older theory; see [2, 3] or [11, Chapter 5.4]. Our result is also valid for ALG3 in the case when there are no cross points. Our current proofs require H^2 -regularity and that $k^3 H^2$ is sufficiently small, i.e. the phase error of the coarse space solution is small enough. (We believe that the H^2 -regularity can be weakened at the expense of a more severe restriction on k and H .) We note that while these types of conditions are meaningful asymptotically, since our results show that the number of iterations will be independent of h , only experiments can tell if the restriction imposed on H makes our results irrelevant for a choice of mesh points that corresponds to a realistic number of mesh points per wave length. We also note that our current theory fails to explain the quite satisfactory performance that we have observed in many of our experiments even without a coarse correction.

The bound for ALG1 is independent of k and h while that of ALG2 deteriorates linearly with the number of points per wave length.

In view of our results and the formulas for the phase error, we have made series of experiments with a fixed $k^3 H^2$ as well as $k^3 h^2$.

5. Numerical Results

The software used was developed with the PETSc library [1] supplied by Argonne National laboratories, as well as Matlab (TM), a product of Mathworks, Inc. We would like to acknowledge the generous help of the PETSc implementors in developing and debugging our code. The platforms for our computations are a Silicon Graphics Reality Monster (TM) parallel computer at Argonne National Laboratories and local workstations.

We now describe the geometry and discretization used in our numerical experiments. In all cases Ω is a unit square discretized with Q_1 elements, and the subregions Ω_j are built from a decomposition of Ω into nonoverlapping square subregions with a layer of δ elements added in all directions. The relevant parameters for describing an experiment are:

- n the number of grid points per side of the square fine mesh and $h = 1/(n-1)$ with Ω a unit square; n_c and h_c the analogs for the coarse mesh;
- k the spatial frequency;
- n_{sub} the number of subregions;
- δ the number of elements across half of the overlap;
- ppw the number of points per wave length on the fine grid; $ppw = \frac{2\pi}{kh}$; ppw_c the analog for the coarse grid.

The number of points per wavelength is a non-dimensional measure of resolution.

We first compare the performance of the three algorithms on a 2×2 set of square subregions without using a coarse grid. Here and below, we say that an algorithm has converged at a given iteration if the ℓ^2 norm of the preconditioned residual is less than 10^{-6} of that of its original value. In some cases we also discuss the relative error at termination of the iteration measured by the ℓ_2 norm of the

difference between the final iterate and the exact solution of the discrete system divided by the ℓ_2 norm of the same exact solution. An iteration number given in parentheses indicates divergence and the iteration number at which the iteration was stopped.

For ALG1 GMRES acceleration had to be used at all times to obtain convergence; moreover, given $n = 97$ or 129 , $\delta = 1, 2, 3, 4$, or 5 , and $ppw = 10$ or 20 , ALG1 never converges in fewer than 40 iterations. Moreover, the relative error in the solution always exceeds 10^{-4} , possibly reflecting ill-conditioning.

For ALG2, we obtain convergence provided we use either overlap or GMRES acceleration. Given the resolution $n = 97$ or 129 and $ppw = 10$, ALG2 diverges when $\delta = 0$ unless GMRES acceleration is applied. Even with acceleration but without overlap, ALG2 appears to be worse than ALG1. With the smallest overlap, $\delta = 1$, ALG2 converges without acceleration. The convergence in about 42 iterations, is similar to that of ALG1, but the error is one tenth of that encountered in ALG1, apparently reflecting better conditioning. With acceleration and $\delta = 1$, ALG2 converges in 13 iterations for both resolutions.

ALG3 does not converge at all without overlap, but with overlap it outperforms both ALG1 and ALG2. Given $ppw = 10$ the resolution $n = 97$ or 129 and $\delta = 1, 2, 3$ or 4 , ALG3 consistently converges at least twice as fast as ALG2. For these parameter values ALG3 always converges in fewer than 10 iterations, and the relative error is always less than 10^{-6} . Given $ppw = 20$, whether $n = 97$ or $n = 129$, the results are the *same* provided that the ratio of δ to ppw is maintained; this indicates that ALG3 has converged with respect to resolution. However, when a coarse grid is used, the ratio of δ to ppw ceases to be an accurate determinant of performance.

In the remaining numerical results we investigate the behavior of ALG3 in the case of many subregions. Generally speaking, in the many subregion case ALG3 needs either GMRES acceleration or a sufficiently fine coarse grid to converge. Table 1 shows the results of several runs with ALG3. For the two sub-tables, $k = 20.11$ and $ppw = 20.00$ (above) and $k = 31.92$ and $ppw=25.20$ (below), and the mesh size is 65×65 (above) and 129×129 (below). The two choices of parameters are related by a constant value of k^3h^2 . In all cases there are 8×8 subregions, and the coarse mesh size varies as indicated in the left column. Within each cell to the left of the double line are presented the parameters n_c (above) and ppw_c (below) for the row; within each cell to the right of the double line are presented data for the unaccelerated algorithm (upper row), the accelerated algorithm (lower row), the iteration count (left column), the normalized residual (right column).

In all the above cases without a coarse grid, GMRES forces convergence in 18 or fewer iterations. By contrast, the algorithm sometimes fails to converge when neither a coarse grid nor GMRES is used. Indeed, other experimental results suggest that with a larger number of subregions ($> 1,000$) convergence without a coarse grid generally requires acceleration.

In particular for a run not shown in our tables with $k=40.21$, $ppw=40.00$, a 257×257 fine grid, no coarse grid, $\delta = 1, 2, 3$, and a 32×32 array of overlapping subregions, the accelerated algorithm converges in 49 to 52 iterations, while the unaccelerated algorithm diverges in two of three cases. For $\delta = 2, 3$ a crude coarse grid correction with $ppw_c=2.66$ grid lowers the iteration count to 28 for the accelerated algorithm, but the unaccelerated algorithm still diverges for $\delta = 1, 2, 3$.

TABLE 1. Tables for **ALG3**

δ	3	2	1	
$n_c=0;$ $ppw_c = 0.00$	(101) 3.87e-01 15 8.97e-07	31 9.90e-07 16 6.84e-07	20 6.91e-07 17 5.75e-07	
$n_c=9;$ $ppw_c = 2.81$	54 9.47e-07 14 5.99e-07	(101) 1.15e+01 14 6.50e-07	(34) 5.85e+01 15 7.05e-07	
$n_c=17;$ $ppw_c = 5.31$	45 9.58e-07 13 9.85e-07	20 9.28e-07 12 7.44e-07	19 9.53e-07 11 4.44e-07	
$n_c=33;$ $ppw_c = 10.31$	42 9.44e-07 13 4.17e-07	21 7.33e-07 12 9.41e-07	17 7.04e-07 11 4.28e-07	

δ	3	2	1	
$n_c=0;$ $ppw_c = 0.00$	18 9.29e-07 15 6.19e-07	21 7.32e-07 16 6.38e-07	24 7.15e-07 18 7.98e-07	
$n_c=17;$ $ppw_c = 3.35$	19 8.09e-07 13 6.60e-07	(101) 5.49e-03 14 9.70e-07	(30) 2.59e+01 17 5.72e-07	
$n_c=33;$ $ppw_c = 6.50$	36 9.12e-07 13 5.41e-07	17 7.98e-07 13 4.86e-07	22 8.42e-07 13 7.81e-07	
$n_c=65;$ $ppw_c = 12.79$	18 9.09e-07 12 6.20e-07	17 6.87e-07 12 6.58e-07	23 8.72e-07 13 9.02e-07	

Thus far, our highest resolution computations use a 385×385 fine grid with a 24×24 array of subregions, $k = 42.54$, and $ppw = 53.33$. Even without a coarse grid, the unaccelerated algorithm converges in 46 to 59 iterations; with GMRES acceleration the algorithm converges in 42 to 44 iterations. With $ppw_c=4.58$ the accelerated algorithm requires 16 to 18 iterations, but the unaccelerated algorithm diverges.

In general, when GMRES acceleration is used we *always* see convergence, even *without* a coarse grid. When a coarse grid is used together with GMRES acceleration, the coarse correction is helpful provided $ppw_c \geq 3$. The success of the GMRES accelerated version of ALG3 evidently comes from the restriction of the spectrum to the right half-plane, which we always observe when using an Arnoldi method to estimate the spectrum. This observation would indicate that ALG3 could be accelerated using less memory intensive techniques such as QMR or generalized conjugate residual acceleration or, even, using Richardson's method with a suitable parameter. Certainly, these are some of the possibilities we will investigate.

References

1. Satish Balay, William Gropp, Lois Curfman McInnes, and Barry F. Smith, *PETSc, the portable, extensible toolkit for scientific computation*, Argonne National Laboratory, 2.0.17 April 5, 1997 ed.
2. Xiao-Chuan Cai and Olof Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Statist. Comput. **13** (1992), no. 1, 243–258.
3. ———, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems*, SIAM J. Numer. Anal. **30** (1993), no. 4, 936–952.
4. Bruno Després, *Méthodes de décomposition de domaine pour les problèmes de propagation d'ondes en régime harmonique*, Ph.D. thesis, Paris IX Dauphine, October 1991.

5. Souad Ghanemi, *Méthode de décomposition de domaine avec conditions de transmissions non locales pour des problèmes de propagation d'ondes*, Ph.D. thesis, Paris IX Dauphine, January 1996.
6. Frank Ihlenburg and Ivo Babuška, *Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation*, Inter. J. Numer. Meth. Eng. **38** (1995), 3745–3774.
7. ———, *Finite element solution of the Helmholtz equation with high wave number, Part I: The h-version of the FEM*, Computers Math. Applic. **30** (1995), no. 9, 9–37.
8. ———, *Finite element solution of the Helmholtz equation with high wave number, Part II: The h-p-version of the FEM*, SIAM J. Numer. Anal. **34** (1997), no. 1, 315–358.
9. L. C. McInnes, R. F. Susan-Resiga, D. E. Keyes, and H. M. Atassi, *Additive Schwarz methods with nonreflecting boundary conditions for the parallel computation of Helmholtz problems*, Tenth International Symposium on Domain Decomposition Methods for Partial Differential Equations (Xiao-Chuan Cai, Charbel Farhat, and Jan Mandel, eds.), AMS, 1997, Submitted.
10. Jens M. Melenk, *On generalized fininte element methods*, Ph.D. thesis, University of Maryland, 1995.
11. Barry F. Smith, Petter E. Bjørstad, and William D. Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.

XIAO-CHUAN CAI, UNIVERSITY OF COLORADO, BOULDER, CO 80309.
URL: [HTTP://WWW.CS.COLORADO.EDU/~CAI](http://WWW.CS.COLORADO.EDU/~CAI).
E-mail address: cai@cs.colorado.edu

MARIO A. CASARIN, IMECC-UNICAMP, CAIXA POSTAL 6065, 13081 - 970 - CAMPINAS - SP, BRAZIL. URL: [HTTP://WWW.IME.UNICAMP.BR/~CASARIN](http://WWW.IME.UNICAMP.BR/~CASARIN).
E-mail address: casarin@ime.unicamp.br

FRANK W. ELLIOTT, JR, COURANT INSTITUTE OF MATHEMATICAL SCIENCES, 251 MERCER STREET, NEW YORK, N.Y. 10012.
E-mail address: elliott@cims.nyu.edu

OLOF B. WIDLUND, COURANT INSTITUTE OF MATHEMATICAL SCIENCES, 251 MERCER STREET, NEW YORK, N.Y. 10012. URL: [HTTP://CS.NYU.EDU/CS/FACULTY/WIDLUND/INDEX.HTML](http://CS.NYU.EDU/CS/FACULTY/WIDLUND/INDEX.HTML).
E-mail address: widlund@cs.nyu.edu

Symmetrized Method with Optimized Second-Order Conditions for the Helmholtz Equation

Philippe Chevalier and Frédéric Nataf

1. Introduction

A schwarz type domain decomposition method for the Helmholtz equation is considered. The interface conditions involve second order tangential derivatives which are optimized (OO2, Optimized Order 2) for a fast convergence. The substructured form of the algorithm is symmetrized so that the symmetric-QMR algorithm can be used as an accelerator of the convergence. Numerical results are shown.

We consider the following type of problem: Find u such that

$$(1) \quad \mathcal{L}(u) = f \text{ in } \Omega$$
$$(2) \quad \mathcal{C}(u) = g \text{ on } \partial\Omega$$

where \mathcal{L} and \mathcal{C} are partial differential operators. We consider Schwarz-type methods for the solving of this problem. The original Schwarz algorithm is based on a decomposition of the domain Ω into overlapping subdomains and the solving of Dirichlet boundary value problems in the subdomains. It has been proposed in [15] to use of more general boundary conditions for the subproblems in order to use a nonoverlapping decomposition of the domain. The convergence speed is also increased dramatically.

More precisely, the computational domain Ω is decomposed into N nonoverlapping subdomains:

$$\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i$$

Let $(\mathcal{B}_{ij})_{1 \leq i,j \leq N}$ be transmission conditions on the interfaces between the subdomains (e.g. Robin BC). What we shall call here a Schwarz type method for the problem (1) is its reformulation: Find $(u_i)_{1 \leq i \leq N}$ such that

$$(3) \quad \mathcal{L}(u_i) = f \text{ in } \Omega_i$$
$$(4) \quad \mathcal{C}(u_i) = g \text{ on } \partial\Omega_i \cap \partial\Omega$$
$$(5) \quad \mathcal{B}_{ij}(u_i) = \mathcal{B}_{ij}(u_j) \text{ on } \partial\Omega_i \cap \partial\Omega_j$$

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35J05.

The above coupled system may be solved iteratively by a Jacobi algorithm:

$$(6) \quad \mathcal{L}(u_i^{n+1}) = f \text{ in } \Omega_i$$

$$(7) \quad \mathcal{C}(u_i^{n+1}) = g \text{ on } \partial\Omega_i \cap \partial\Omega$$

$$(8) \quad \mathcal{B}_{ij}(u_i^{n+1}) = \mathcal{B}_{ij}(u_j^n) \text{ on } \partial\Omega_i \cap \partial\Omega_j$$

It is also possible, and indeed preferable, to write the resulting linear system in the interface unknowns $H = (\mathcal{B}_{ij}(u_i))_{1 \leq i,j \leq N}$ (see [18])

$$(9) \quad \mathcal{A}H = G$$

(G is some computable right hand side) and to solve it by conjugate gradient type methods.

Let us focus first on the interface conditions \mathcal{B}_{ij} . The convergence speed of Schwarz-type domain decomposition methods is very sensitive to the choice of these transmission conditions. The use of exact artificial (also called absorbing boundary conditions) boundary conditions as interface conditions leads to an optimal number of iterations, see [12, 18, 11]. Indeed, for a domain decomposed into N strips, the number of iterations is N , see [18]. Nevertheless, this approach has some drawbacks:

1. the explicit form of these boundary conditions is known only for constant coefficient operators and simple geometries.
2. These boundary conditions are pseudodifferential. The cost per iteration is high since the corresponding discretization matrix is not sparse for the unknowns on the boundaries of the subdomains.

For this reason, it is usually preferred to use partial differential approximations to the exact absorbing boundary conditions. This approximation problem is classical in the field of computation on unbounded domains since the seminal paper of Engquist and Majda [6]. The approximations correspond to “low frequency” approximations of the exact absorbing boundary conditions. In domain decomposition methods, many authors have used them for wave propagation problems [3, 5, 14, 4, 2, 16, 1, 19] and in fluid dynamics [17, 21, 10, 9]. Instead of using “low frequency” approximations to the exact absorbing boundary conditions, it has been proposed to design approximations which minimize the convergence rate of the algorithm, see [22]. These approximations are quite different from the “low frequency” approximations and increase dramatically the convergence speed of the method, see [13] for a convection-diffusion equation. But, in the case of the Helmholtz equation, the same optimization procedure cannot be done, see Sec. 4 below. This is related to the existence of both propagative and evanescent modes for the solution of the Helmholtz equation. Roughly speaking, we will choose the interface conditions in Sec. 4 so that the convergence rate is small for both modes. In a different manner, in [20] overlapping decompositions are considered. In the overlapping regions, the partial differential equation is modified smoothly.

Another important factor is the choice of the linear solver (CG, GMRES, BICG, QMR, etc ...) used to solve the substructured linear system (9). For such methods, the positivity and symmetry of the linear system are important issues [7]. Since the Helmholtz operator is not positive, it is unlikely that \mathcal{A} is positive (and indeed, it is not). But, the Helmholtz operator is symmetric while \mathcal{A} is not. By a change of unknown on H , we shall rewrite (9) such as to obtain a symmetric formulation. It enables us to use the symmetric-QMR algorithm described in [8].

2. Substructuring

In this section, we write the explicit form of the substructured linear problem (9) for the problem below. This section is classical. We want to solve the Helmholtz equation: Find u such that

$$(10) \quad \begin{aligned} (\Delta + \omega^2)(u) &= f \text{ in } \Omega =]0, L_x[\times]0, L_y[\\ \left(\frac{\partial}{\partial n} + i\omega\right)(u) &= g \text{ on } \{0\} \times]0, L_y[\cup\{L_x\} \times]0, L_y[\\ u &= 0 \text{ on }]0, L_x[\times\{0\} \cup]0, L_x[\times\{L_y\} \end{aligned}$$

where $i^2 = -1$, $\frac{\partial}{\partial n}$ is the outward normal derivative and ω is positive. The domain is decomposed into N nonoverlapping vertical strips:

$$\Omega_k =]L_{k-1,k}, L_{k,k+1}[\times]0, L_y[, \quad 1 \leq k \leq N$$

with $L_{0,1} = 0 < L_{1,2} < \dots < L_{N,N+1} = L_x$. The transmission condition on the interfaces is of the form $\mathcal{B}_k = \frac{\partial}{\partial n_k} + i\omega - \eta \frac{\partial^2}{\partial \tau_k^2}$ where $\frac{\partial}{\partial \tau_k}$ is the tangential derivative on the boundary of $\partial\Omega_k$ and $\eta \in \mathbb{C}$ will be chosen in the section 4.

Problem (10) is equivalent to: Find $(u_k)_{1 \leq k \leq N}$ such that

$$(11) \quad \begin{aligned} (\Delta + \omega^2)(u_k) &= f \text{ in } \Omega_k \\ \mathcal{B}_k(u_k) &= \mathcal{B}_k(u_{k-1}) \text{ on } \partial\Omega_k \cap \partial\Omega_{k-1} \\ \mathcal{B}_k(u_k) &= \mathcal{B}_k(u_{k+1}) \text{ on } \partial\Omega_k \cap \partial\Omega_{k+1} \\ \left(\frac{\partial}{\partial n} + i\omega\right)(u_k) &= g \text{ on } (\{0\} \times]0, L_y[\cup\{L_x\} \times]0, L_y[) \cap \partial\Omega_k \\ u_k &= 0 \text{ on } (]0, L_x[\times\{0\} \cup]0, L_x[\times\{L_y\}) \cap \partial\Omega_k \end{aligned}$$

The continuity of the solution and its derivative on $\Omega_k \cap \partial\Omega_{k+1}$ are ensured by the interface conditions \mathcal{B}_k and \mathcal{B}_{k+1} . Let $\Sigma_{k,k+1} = \partial\Omega_k \cap \partial\Omega_{k+1}$, $1 \leq k \leq N$. Let us define now the vector of unknowns

$$H = (\mathcal{B}_1(u_1)|_{\Sigma_{1,2}}, \dots, \mathcal{B}_k(u_k)|_{\Sigma_{k-1,k}}, \mathcal{B}_k(u_k)|_{\Sigma_{k,k+1}}, \dots, \mathcal{B}_N(u_N)|_{\Sigma_{N-1,N}})$$

Let Π be the interchange operator on the interfaces, on each interface we have:

$$(12) \quad \Pi(0, \dots, 0, h_k^{k-1}, h_k^{k+1}, 0, \dots, 0) = (0, \dots, 0, h_{k-1}^k, h_{k+1}^k, 0, \dots, 0)$$

Let T' be the linear operator defined by:

$$(13) \quad \begin{aligned} T'(h_1^2, \dots, h_k^{k-1}, h_k^{k+1}, \dots, h_N^{N-1}, f, g) &= \\ (\tilde{\mathcal{B}}_1(v_1)|_{\Sigma_{1,2}}, \dots, \tilde{\mathcal{B}}_k(v_k)|_{\Sigma_{k-1,k}}, \tilde{\mathcal{B}}_k(v_k)|_{\Sigma_{k,k+1}}, \dots, \tilde{\mathcal{B}}_N(v_N)|_{\Sigma_{N-1,N}}) \end{aligned}$$

where $\tilde{\mathcal{B}}_k = -\frac{\partial}{\partial n_k} + i\omega - \eta \frac{\partial^2}{\partial \tau_k^2}$ and v_k satisfies:

$$(14) \quad (\Delta + \omega^2)(v_k) = f \text{ in } \Omega_k$$

$$(15) \quad \mathcal{B}_k(v_k) = h_k^{k-1} \text{ on } \partial\Omega_k \cap \partial\Omega_{k-1}$$

$$(16) \quad \mathcal{B}_k(v_k) = h_k^{k+1} \text{ on } \partial\Omega_k \cap \partial\Omega_{k+1}$$

$$(17) \quad \begin{aligned} \left(\frac{\partial}{\partial n} + i\omega\right)(v_k) &= g \text{ on } (\{0\} \times]0, L_y[\cup\{L_x\} \times]0, L_y[) \cap \partial\Omega_k \\ v_k &= 0 \text{ on } (]0, L_x[\times\{0\} \cup]0, L_x[\times\{L_y\}) \cap \partial\Omega_k \end{aligned}$$

It can be shown that

LEMMA 1. *Finding $(u_k)_{1 \leq k \leq N}$ solution to (11) is equivalent to finding $H = (h_1^2, \dots, h_k^{k-1}, h_k^{k+1}, \dots, h_N^{N-1})$ such that*

$$(18) \quad (Id - \Pi T)(H) = \Pi T'(0, 0, \dots, 0, f, g) \equiv G$$

where

$$(19) \quad T = T'(\cdot, \cdot, \dots, \cdot, 0, 0).$$

3. Symmetrization of the substructured system

The linear system (18) may be solved by a relaxed Jacobi algorithm

$$(20) \quad H^{n+1} = \theta \Pi T(H^n) + (1 - \theta)H^n + G, \quad \theta \in (0, 2)$$

It is more efficient to use linear solvers based on Krylov subspaces such as GMRES, BICGSTAB or QMR. As mentioned in the introduction, the convergence rate of such methods is very sensitive to the positivity and symmetry of the linear system. Due to the non-positivity of the Helmholtz operator, it is unlikely that the substructured problem (18) be positive. Nevertheless, the Helmholtz equation is symmetric. It seems thus possible to obtain a symmetric reformulation of (18). For this, we need

DEFINITION 2. Let

$$W = \prod_{k=2}^{N-1} (L^2([0, L_y]) \times L^2([0, L_y]))$$

be the space of complex valued traces on the interfaces. An element $H \in W$ is denoted

$$H = (h_1^2, h_2^1, \dots, h_k^{k-1}, h_k^{k+1}, \dots, h_{N-1}^N, h_N^{N-1})$$

The space W is endowed with a bilinear operator from $W \times W$ to \mathbb{C}

$$\forall G, H \in W, \quad (G, H)_b = \sum_{k=2}^{N-1} \int_0^{L_y} (g_k^{k-1} h_k^{k-1} + g_k^{k+1} h_k^{k+1})$$

Let A be a linear operator from W to W . The transpose of A , denoted A^T is the linear operator from W to W such that

$$\forall G, H \in W, \quad (AG, H)_b = (G, A^T H)_b$$

An operator A is symmetric iff $A^T = A$.

The notion of symmetry for complex valued linear operator corresponds to the notion of complex symmetric linear systems studied in [8]. It is different from the notion of Hermitian operators since we have no conjugation. We have

LEMMA 3. *Let Π be defined in (12). Then,*

$$\Pi^T = \Pi; \Pi^2 = Id$$

and the operator Π admits symmetric square roots. Let $\Pi^{1/2}$ denote one of these. Let T be defined in (19). Then,

$$(21) \quad T^T = T$$

PROOF. As for Π , it is enough to consider one interface. The operators Π and $\Pi^{1/2}$ may be represented by the matrices

$$\Pi = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \Pi^{1/2} = \begin{pmatrix} 1 + \frac{i}{2} & 1 - \frac{i}{2} \\ 1 - \frac{i}{2} & 1 + \frac{i}{2} \end{pmatrix}$$

As for T the result comes from integration by parts of the Helmholtz operator. \square

This enables us to propose the following symmetrization:

LEMMA 4. *Problem (18) is equivalent to: Find $H \in W$ such that*

$$(22) \quad (Id - \Pi^{1/2} T \Pi^{1/2})(\Pi^{-1/2} H) = \Pi^{-1/2} G$$

Moreover

$$(Id - \Pi^{1/2} T \Pi^{1/2})^T = (Id - \Pi^{1/2} T \Pi^{1/2})$$

and the operators $(Id - \Pi^{1/2} T \Pi^{1/2})$ and $(Id - \Pi T)$ have the same set of eigenvalues.

Other possibilities for symmetrizing (18) are

$$(\Pi - T)(H) = \Pi(G) \text{ or } (T - \Pi T)(H) = T(G)$$

They don't have the same eigenvalues as the original formulation and thus lead to slower convergence, see Figure 2.

4. Optimization of the interface conditions

The optimization is performed on the parameter η . The more $(Id - \Pi T)$ is close to the identity, the better the convergence of any iterative method will be. It seems then natural to minimize the norm of ΠT . Since Π is an isometry and does not depend on η , this is equivalent to minimize the norm of T . But we have

PROPOSITION 5. Let the plane \mathbb{R}^2 be decomposed into two half-spaces $\Omega_1 =]-\infty, 0[\times \mathbb{R}$ and $\Omega_2 =]0, \infty[\times \mathbb{R}$. For all $\eta \in \mathbb{C}$,

$$\|T\| = 1$$

Indeed, a simple Fourier analysis shows that the operator $T(., ., 0, 0)$ may be represented as an operator valued matrix:

$$T = \begin{pmatrix} R & 0 \\ 0 & R \end{pmatrix}$$

where R is a pseudo-differential operator whose symbol is

$$(23) \quad \rho(k, \eta) = \frac{i\sqrt{\omega^2 - k^2} - i\omega - \eta k^2}{i\sqrt{\omega^2 - k^2} + i\omega + \eta k^2}$$

where k is the dual variable of y for the Fourier transform. For $k = \omega$, $\rho(\omega, \eta) = 1$ for all $\eta \in \mathbb{C}$.

It is thus impossible to optimize the spectral radius of $T(., ., 0, 0)$ as is done in [13].

Nevertheless, for any value of η , $|\rho(0, \eta)| = 0$. This indicates that the convergence rate should be good for low frequencies as it is already the case if $\eta = 0$, see [3]. It seems thus interesting to use η for having a good convergence rate also for high frequencies. From (23), it is then enough to consider η real. The choice of η is

TABLE 1. Influence of the number of subdomains

Number of subdomains	2	3	4	6	10	15	20
Relaxed Jacobi + Robin BC = 0.5	50	62	75	130	216	406	832
QMR + Optimized BC	10	18	24	34	59	92	126

TABLE 2. Nbr of iterations vs. nbr of points per wavelength

	10 pts / lo	20 pts / lo	40 pts / lo
Relaxed Jacobi + Robin BC=0.5	130	195	299
QMR + Optimized BC	34	38	41

TABLE 3. Influence of ω

	$\omega = 6$	$\omega = 20$	$\omega = 60$
Relaxed Jacobi + Robin BC	130	155	664
QMR + Optimized BC	34	50	60

done by minimizing the integral of $|\rho(k, \eta)|$ over the evanescent modes “admissible” on the computational grid:

$$(24) \quad \min_{\eta \in \mathbb{R}} \int_{\omega}^{k_{max}} |\rho(k, \eta)| dk$$

where $k_{max} \simeq 1/h$ and h is the typical mesh size. This is all the more important that if $\eta = 0$, $|\rho(k, 0)| = 1$ as soon as $|k| > \omega$.

5. Numerical results

The Helmholtz equation was discretized by a 5-point finite difference scheme. Except for Table 3, the number of points per wavelength was between 20 and 25. The problems in the subdomains were solved exactly by LU factorization. As for the substructured problem, we compared different iterative methods: relaxed Jacobi algorithm and the symmetric-QMR method. The symmetric-QMR algorithm was chosen since it is adapted to complex symmetric linear systems. We also tested various interface conditions:

$$\begin{aligned} \mathcal{B} &= \partial_n + i\omega \text{ (Robin BC)} \\ \mathcal{B} &= \partial_n + i\omega + \frac{i}{2\omega} \partial_{\tau}^2 \text{ (Order 2 BC)} \\ \mathcal{B} &= \partial_n + i\omega - \eta \partial_{\tau}^2 \text{ (optimized real } \eta, \text{ cf (24))} \\ \mathcal{B} &= \partial_n + i\omega + \left(\frac{i}{2\omega} - \eta\right) \partial_{\tau}^2 \text{ (Order 2 + optimized real } \eta, \text{ } \eta \text{ as above)} \end{aligned}$$

The stopping criterion was on the maximum of the error between the converged numeric solution and the approximation generated by the algorithm, $\|e\|_{\infty} < 10^{-5}$. Figure 1 ($\omega = 6 \iff$ three wavelengths per subdomain), Table 1, Table 2 and Table 3 enable to compare the proposed method with a relaxed Jacobi algorithm.

In Figure 2, only the formulation of the substructured problem is varied while the interface conditions are the same. Table 4 shows the importance of the interface conditions. We have used Jacobi’s algorithm with different interface conditions.

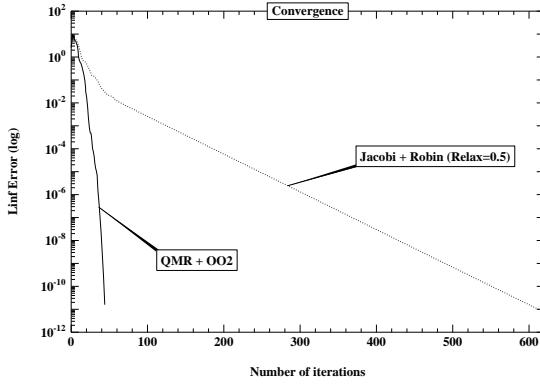


FIGURE 1. Comparison of the proposed method with a Jacobi-Robin algorithm

TABLE 4. Jacobi's algorithm - Influence of the interface conditions

	Robin	Order 2	η real opt	η real opt + order 2
Nbr of iterations	195	> 1000	140	51
Best relaxation's coefficient	0.5	any	0.6	0.9

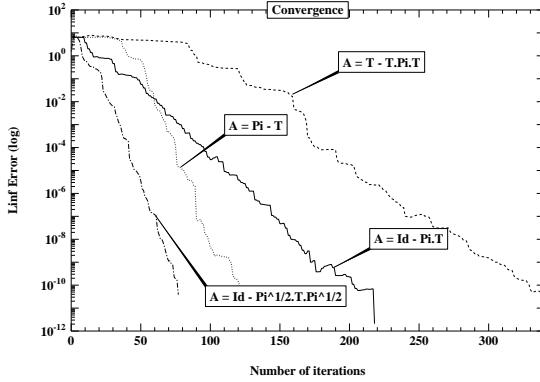


FIGURE 2. Influence of the substructured formulation

6. Perspectives

We have presented a domain decomposition method for the Helmholtz equation and a decomposition of the domain into strips. The present method has already been extended to the Maxwell operator. We shall consider an arbitrary decomposition of the domain in a forthcoming paper.

References

1. A. de la Bourdonnaye, C. Fahrat, A. Macedo, F. Magoulès, and F.X. Roux, *A nonoverlapping domain decomposition method for the exterior Helmholtz problem*, DD10 Proceedings, 1997.

2. M. Casarin, F. Elliot, and O. Windlund, *An overlapping Schwarz algorithm for solving the Helmholtz equation*, DD10 Proceedings, 1997.
3. B. Després, *Domain decomposition method and the Helmholtz problem. II*, Mathematical and numerical aspects of wave propagation. Proceedings of the 2nd international conference held (Kleinman Ralph eds. and al., eds.), SIAM, June 7-10 1993, pp. 197–206.
4. B. Després and J.D. Benamou, *A domain decomposition method for the Helmholtz equation and related optimal control problems*, J. Comp. Phys. **136** (1997), 68–82.
5. B. Despres, P. Joly, and J.E. Roberts, *Domain decomposition method for the harmonic Maxwell equations*, International Symposium on Iterative methods in linear algebra, 1991, pp. 475–484.
6. B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp. **31** (1977), no. 139, 629–651.
7. R.A Freund, G. Golub, and Nachtigal, *Iterative solution of linear systems*, Acta Numerica (1992), 57–100.
8. R.W. Freund, *Conjuguate gradient-type methods for linear systems with complex symmetric coefficient matrices*, J. Sci. Stat. Comput. **13** (1992), no. 1, 425–448.
9. M. Garbey, *A Schwarz alternating procedure for singular perturbation problems*, SIAM J. Sci. Comput. **17** (1996), no. 5, 1175–1201.
10. F. Gastaldi, L. Gastaldi, and A. Quarteroni, *Adaptative domain decomposition methods for advection dominated equations*, East-West J. Numer. Math. **4** (1996), no. 3, 165–206.
11. S. Ghanemi, P. Joly, and F. Collino, *Domain decomposition method for harmonic wave equations*, Third international conference on mathematical and numerical aspect of wave propagation, 1995, pp. 663–672.
12. T. Hagstrom, R.P. Tewarson, and A.Jazcilevich, *Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems*, Appl. Math. Lett. **1** (1988), no. 3, 299–302.
13. C. Japhet, *Optimized Krylov-Ventcell method. application to convection-diffusion problems*, DD9 Proceedings, John Wiley & Sons Ltd, 1996.
14. B. Lichtenberg, B. Webb, D. Meade, and A.F. Peterson, *Comparison of two-dimensional conformal local radiation boundary conditions*, Electromagnetics **16** (1996), 359–384.
15. P.L. Lions, *On the Schwarz alternating method III: A variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, 1989, pp. 202–223.
16. L.C. McInnes, R.F. Susan-Resiga, D.E. Keyes, and H.M. Atassi, *Additive Schwarz methods with nonreflecting boundary conditions for the parallel computation of Helmholtz problems*, These proceedings.
17. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, M³AS **5** (1995), no. 1, 67–93.
18. F. Nataf, F. Rogier, and E. de Sturler, *Domain decomposition methods for fluid dynamics, Navier-Stokes Equations and Related Nonlinear Analysis* (A. Sequeira, ed.), Plenum Press Corporation, 1995, pp. 367–376.
19. B. Stupfel, *A fast-domain decomposition method for the solution of electromagnetic scattering by large objects*, IEEE Transactions on Antennas and Propagation **44** (1996), no. 10, 1375–1385.
20. Sun, Huosheng, Tang, and Wei-Pai, *An overdetermined Schwarz alternating method*, SIAM J. Sci. Comput. **17** (1996), no. 4, 884–905.
21. P. Le Tallec and T. Sassi, *Domain decomposition with non matching grids*, Tech. report, INRIA, 1991.
22. K.H. Tan and M.J.A. Borsboom, *On generalized Schwarz coupling applied to advection-dominated problems*, Domain decomposition methods in scientific and engineering computing. Proceedings of the 7th international conference on domain decomposition, vol. 180, AMS Contemp. Math., 1994, pp. 125–130.

THOMSON-CSF LABORATOIRE CENTRAL DE RECHERCHE, 91404 ORSAY CEDEX, FRANCE
E-mail address: chevalie@thomson-lcr.fr - chevalie@cmapx.polytechnique.fr

C.M.A.P., CNRS UMR 7641, ECOLE POLYTECHNIQUE, 91129 PALAISEAU CEDEX, FRANCE
E-mail address: nataf@cmapx.polytechnique.fr

Non-overlapping Schwarz Method for Systems of First Order Equations

Sébastien Clerc

1. Introduction

Implicit time-stepping is often necessary for the simulation of compressible fluid dynamics equations. For slow transient or steady-state computations, the CFL stability condition of explicit schemes is indeed too stringent. However, one must solve at each implicit time step a large linear system, which is generally unsymmetric and ill-conditioned. In this context, domain decomposition can be used to build efficient preconditioners suited for parallel computers. This goal was achieved by [7, 9], among others.

Still, important theoretical questions remain open to our knowledge, such as: estimate of the condition number, optimality of the preconditioner. This aspect contrasts with the existing results in structural mechanics, where the subspace correction framework allows a complete analysis (see for instance [15] and the references therein). This work is a preliminary step towards a better understanding of domain decomposition for systems of equations in the context of fluid dynamics.

To this purpose, we study the steady linearized equations, following the ideas of [6] and [11] for instance. In section 2, we present the derivation of these equations from the time-dependent non-linear hyperbolic systems of conservation laws. Symmetrization is also addressed, as well as the nature of the resulting equations.

A classical well-posedness result is recalled in section 3 for the boundary value problem. An energy estimate allows one to prove the convergence of the Schwarz iterative method, using the same arguments as [4] for the Helmholtz problem. This result generalizes those of [6] for scalar transport equations, and [12] for one-dimensional systems.

In section 4, we emphasize the difference between the purely hyperbolic case and the elliptic case, as far as the convergence of the Schwarz method is concerned. We also mention the influence of the transmission condition on the convergence for a model problem.

Section 5 deals with the space-discretization of the problem with a finite volume method and a first order upwind scheme. We propose two different implementations: a direct one which can be interpreted as a block-Jacobi preconditioner, and a Schur complement formulation. The latter is all the more useful when used in

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35L65.

combination with the GMRES [14] algorithm. Numerical results are given for the Cauchy-Riemann equations and Saint-Venant's equations of shallow water flow.

Finally, conclusions are drawn in section 6.

2. Derivation of the equations

The equations of compressible fluid dynamics are generally formulated as a non-linear system of conservation laws:

$$(1) \quad \partial_t u + \sum_{\alpha=1}^d \partial_\alpha F^\alpha(u) = 0.$$

Here the solution u is a vector of \mathbb{R}^p and the fluxes F^α are non-linear vector-valued functions of u . The equations are posed in \mathbb{R}^d and the index α refers to the direction in the physical space.

Suppose that the solution $u^n(x)$ at time t^n is known. We seek a solution u^{n+1} at time $t^{n+1} = t^n + \delta t$. If the increment $\delta u = u^{n+1} - u^n$ is small enough, we can write the following linearized implicit equation:

$$\delta u / \delta t + \sum \partial_\alpha (D_u F^\alpha(u^n) \delta u) = - \sum \partial_\alpha F^\alpha(u^n),$$

$D_u F^\alpha$ being the Jacobian matrix of the flux F^α .

The next step consists in writing the non-conservative form, which is valid if u^n is smooth:

$$(2) \quad \left[(1/\delta t) Id + \sum \partial_\alpha D_u F^\alpha(u^n) \right] \delta u + \sum D_u F^\alpha \partial_\alpha \delta u = - \sum \partial_\alpha F^\alpha(u^n).$$

Finally, we recall that these equations can be symmetrized if the system of conservation laws (1) admits a mathematical entropy S (see for instance [8]). We multiply system (2) by the Hessian matrix $D_{uu} S$ of the entropy. The resulting system takes the following form:

$$(3) \quad A^0 u + \sum_{\alpha=1}^d A^\alpha \partial_\alpha u = f,$$

with:

$$(4) \quad A^0 = D_{uu} S(u^n) \left[(1/\delta t) Id + \sum \partial_\alpha D_u F^\alpha(u^n) \right],$$

$$A^\alpha = D_{uu} S(u^n) \cdot D_u F^\alpha(u^n).$$

The matrices A^α are symmetric, but A^0 may be any matrix for the moment.

In the sequel, we will consider boundary value problems for general symmetric systems of first order equations of type (3).

When going from the time-dependent system (1) to the steady system (3), the equations may not remain hyperbolic. The scalar case studied by [6, 17] and the one-dimensional case studied by [12] are two examples of hyperbolic steady equations.

The linearized steady Euler equations in 2-D are also hyperbolic in the super-sonic regime, but become partially elliptic in the subsonic regime (see for instance

[16]). This aspect dramatically changes the nature of the boundary value problem. The simplest model exhibiting such behaviour is the time-dependent Cauchy-Riemann system:

$$(5) \quad \begin{cases} \partial_t u - \partial_x u + \partial_y v = 0 \\ \partial_t v + \partial_x v + \partial_y u = 0, \end{cases}$$

which becomes purely elliptic after an implicit time-discretization. This system therefore retains some of the difficulties involved in the computation of subsonic flows.

3. A well-posed boundary value problem

Let Ω be a domain of \mathbb{R}^d with a smooth boundary $\partial\Omega$. If $n = (n_1, \dots, n_d)$ is the outward normal vector at $x \in \partial\Omega$, we denote by $A_n = \sum A^\alpha n_\alpha$ the matrix of the flux in the direction n . This matrix is real and symmetric, and therefore admits a complete set of real eigenvectors. We define $A_n^+ = P^{-1}\Lambda^+P$ with $\Lambda^+ = \text{diag}(\max\{\lambda_i, 0\})$. Similarly, $A_n^- = P^{-1}\Lambda^-P$ with $\Lambda^- = \text{diag}(\min\{\lambda_i, 0\})$, so that $A_n = A_n^+ + A_n^-$.

Next, we introduce a minimal rank positive-negative decomposition of the matrix A_n . Let A_n^{pos} (respectively A_n^{neg}) be a symmetric positive (resp. negative) matrix such that $A_n = A_n^{pos} + A_n^{neg}$ and $\text{rank}(A_n^{pos}) = \text{rank}(A_n^+)$, $\text{rank}(A_n^{neg}) = \text{rank}(A_n^-)$. The simplest choice is of course $A_n^{pos} = A_n^+$ and $A_n^{neg} = A_n^-$: it corresponds to a decomposition with local characteristic variables, which is also the first order absorbing boundary condition (cf. [5]). In the scalar case ($p = 1$), $A_n^{neg} = A_n^-$ is the only possible choice.

With these notations, we can define a dissipative boundary condition of the form:

$$(6) \quad A_n^{neg}u = A_n^{neg}g \quad \text{on } \partial\Omega.$$

In the scalar case, this condition amounts to prescribing the boundary data only on the inflow boundary.

With some regularity and positivity assumptions, one can prove the following theorem:

THEOREM 1. *Let $f \in L^2(\Omega)^p$ and g such that $\int_{\partial\Omega} A_n^{neg}g \cdot g < \infty$ be given. There exists a unique solution $u \in L^2(\Omega)^p$, with $\sum A^\alpha \partial_\alpha u \in L^2(\Omega)^p$, such that*

$$(7) \quad \begin{cases} A^0 u + \sum A^\alpha \partial_\alpha u = f & \text{in } \Omega \\ A_n^{neg}u = A_n^{neg}g & \text{on } \partial\Omega, \end{cases}$$

The solution satisfies the following estimate:

$$(8) \quad C_0 \|U_k\|_{L^2}^2 + \int_{\partial\Omega} A_n^{pos} U_k \cdot U_k \leq \frac{1}{C_0} \|f\|_{L^2}^2 - \int_{\partial\Omega} A_n^{neg} g \cdot g.$$

Proof: We refer to [10] and [1] for the proof in the case where $g = 0$. See also [6] for the scalar case. The general case ($g \neq 0$) is addressed in [3].

4. The Schwarz algorithm

For simplicity, we consider a non-overlapping decomposition of the domain Ω : $\bigcup_{1 \leq i \leq N} \Omega_i = \Omega$. We will denote by $\Gamma_{i,j} = \partial\Omega_i \cap \partial\Omega_j$ the interface between two subdomains, when it exists. If n is the normal vector to $\Gamma_{i,j}$, oriented from Ω_i to

Ω_j , we set $A_n = A_{i,j}$. Hence, $A_{i,j} = -A_{j,i}$. We can decompose the global problem in Ω in a set of local problems:

$$\begin{cases} A^0 u_i + \sum A^\alpha \partial_\alpha u_i &= f \quad \text{in } \Omega_i \\ A_n^{neg} u_i &= A_n^{neg} g \quad \text{on } \partial\Omega \cup \partial\Omega_i, \end{cases}$$

which we supplement with the transmission conditions:

$$A_{i,j}^{neg} u_i = A_{i,j}^{neg} u_j \quad A_{j,i}^{neg} u_i = A_{j,i}^{neg} u_j \quad \text{on } \Gamma_{i,j}.$$

We now describe the classical Schwarz algorithm for the solution of these transmission conditions. We make use a vector $U^k = (u_1^k, \dots, u_N^k)$ of local solutions. Let U^0 be given. If U^k is known, U^{k+1} is defined by:

$$(9) \quad \begin{cases} A^0 u_i^{k+1} + \sum A^\alpha \partial_\alpha u_i^{k+1} &= f \quad \text{in } \Omega_i \\ A_n^{neg} u_i^{k+1} &= A_n^{neg} g \quad \text{on } \partial\Omega \cap \partial\Omega_i, \\ A_{i,j}^{neg} u_i^{k+1} &= A_{i,j}^{neg} u_j^k \quad \text{on } \Gamma_{i,j}, \end{cases}$$

Inequality (8) shows that the trace of the solution u_j^k satisfies

$$\sum_{\partial\Omega_j} A_n^{pos} u_j \cdot u_j < \infty.$$

The trace of u_j^k on $\Gamma_{i,j}$ can therefore be used as a boundary condition for u_i^{k+1} , which ensures that the algorithm is well defined.

The Schwarz method converges if each u_i^k tends to the restriction of u to Ω_i as k tends to infinity. More precisely, we can prove the following theorem:

THEOREM 2. *The algorithm (9) converges in the following sense:*

$$\|e_i^k\|_{L^2} \rightarrow 0, \quad \left\| \sum A^\alpha \partial_\alpha e_i^k \right\|_{L^2} \rightarrow 0,$$

where $e_i^k = u - u_i^k$ is the error in subdomain Ω_i .

Proof: The proof is similar to [4] (see also [11]). See [3] for more details.

5. Examples

5.1. Example 1: the scalar case. In the case of decomposition in successive slabs following the flow (see Fig. 1), it is easily seen that the Schwarz method converges in a finite number of steps. The Schwarz method is thus optimal in this case but the parallelization is useless. Indeed, the residual does not decrease until the last iteration: the sequential multiplicative algorithm would be as efficient in this case. This peculiar behaviour is due to the hyperbolic nature of the equations and would also occur for the linearized Euler equations in one dimension and for 2-D supersonic flows.

5.2. Example 2: the Cauchy-Riemann equations. Here the convergence is not optimal but the error is reduced at each iteration. In the case where the domain \mathbb{R}^2 is decomposed in two half-planes, the convergence of the Schwarz method can be investigated with a Fourier transform in the direction of the interface, see [3]. This computation shows that the Schwarz method behaves similarly for this first order elliptic system as for a usual scalar second order elliptic equation. It is possible to define an analogue of the Steklov-Poincare operator in this case, which is naturally non-local.

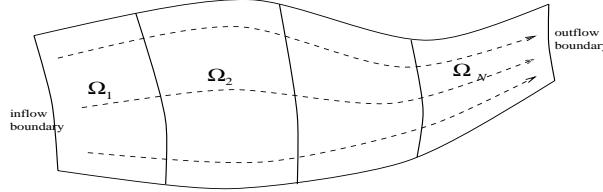


FIGURE 1. Decomposition of the domain for a transport equation.
In this case, convergence is reached in exactly N steps, where N is the number of subdomains.

As mentionned in section 3, any decomposition $A = A^{neg} + A^{pos}$ with minimal rank leads to a dissipative boundary condition $A^{neg}u = A^{neg}g$. For the Cauchy-Riemann equations (5), there is a one-parameter family of such decompositions. In the case where $n = (1, 0)$ for instance, we may take:

$$A^{neg} = \begin{pmatrix} -\cosh^2 \varphi & -\sinh \varphi \cosh \varphi \\ -\sinh \varphi \cosh \varphi & -\sinh^2 \varphi \end{pmatrix},$$

for any real number φ . The computation of the spectrum of the Schwarz method in this case shows that the convergence is optimal when $\varphi = 0$, i.e. when $A^{neg} = A^-$. This case corresponds to the first order absorbing boundary condition [5]. The optimality of this kind of transmission conditions has been first recognized by [11] in the context of convection-diffusion equations.

6. Numerical results with a first order implicit finite volume scheme

6.1. Finite volume discretization. We consider a triangulation of Ω . For simplicity, we will assume that $\Omega \subset \mathbb{R}^2$, but the extension to \mathbb{R}^3 is straightforward. If K is a cell of the triangulation, the set of its edges e will be denoted by ∂K . $|K|$ is the total area of the cell and $|e|$ the length of edge e .

We seek a piecewise constant approximation to (3), u_K being the cell-average of u in cell K . The finite volume scheme reads:

$$(10) \quad |K|A_0u_K + \sum_{e \in \partial K} |e| \Phi_e^K = |K|f_K.$$

In problems arising from a non-linear system of conservation laws, we use the implicit version of Roe's scheme [13], written in the usual non-conservative form:

$$(11) \quad \Phi_e^K(u_K, u_J) = A_n^-[u_J - u_K], \quad \Phi_e^J(u_K, u_J) = A_n^+[u_J - u_K],$$

if e is the common edge between K and J , with a normal vector n oriented from K to J . The averaged Jacobian matrix A_n is computed from the preceding time-step and must satisfy Roe's condition:

$$A_n(u_K, u_J)[u_J - u_K] = [F_n(u_J) - F_n(u_K)].$$

When this condition holds, the implicit scheme (11) is conservative at steady-state.

Note that in the linear case with constant coefficients, (11) is equivalent to the classical first order upwind scheme:

$$\Phi_e^K(u_K, u_J) = -\Phi_e^J = A_n^-u_J + A_n^+u_K.$$

6.2. Boundary conditions. The boundary conditions defined by local characteristics decomposition, namely $A_n^- u = A_n^- g$ are naturally discretized by:

$$\Phi_e = \Phi(u_K, g_e) = A_n^- [g_e - u_K],$$

if e is an edge of cell K lying on the boundary. For general dissipative boundary conditions of the type $A_n^{neg} u = A_n^{neg} g$, one can use a different scheme at the boundary: this modification is linked to the so-called preconditioning technique, see [16, 3] for more details. An alternative will be described in the sequel.

6.3. The Schwarz method. It is easily seen that the Schwarz method, discretized by a first order upwind scheme can be interpreted as a block-Jacobi solver for the global linear system. The local problems in each subdomain leads to a local sub-system, which can be solved with a *LU* factorization of the corresponding matrix block.

Alternatively, a substructuring approach is possible. In finite element applications, this approach consists in eliminating all interior unknowns and writing a condensed system involving only the unknowns at the interfaces. The resulting matrix is the so-called Schur complement. The dimension of the condensed linear system is much lower than that of the initial system. This is especially interesting with GMRES, as all intermediate vectors of the Krylov subspace have to be stored.

However, the unknowns at the interfaces do not appear explicitly in the finite volume discretization. Rather, they are reconstructed from the interior values. To bypass this difficulty, we propose to introduce redundant unknowns at the interface. Namely, if e is an edge lying on an interface, we define u_e by:

$$(12) \quad A_n^- u_e = A_n^- u_K, \quad A_n^+ u_e = A_n^+ u_J.$$

The flux at the interface is thus:

$$\Phi_e^K = A_n^- [u_e - u_K], \quad \Phi_e^J = A_n^+ [u_J - u_e].$$

Thanks to this formulation, we can solve all the interior unknowns u_K in terms of the redundant unknowns u_e .

With the latter formulation, one can easily implement transmission conditions of the type $A_n^{neg} u = A_n^{neg} g$: the definition of the interface-based unknowns simply becomes:

$$A_n^{neg} u_e = A_n^{neg} u_K, \quad A_n^{pos} u_e = A_n^{pos} u_J.$$

This formulation has been used for the Cauchy-Riemann equations to verify the results of the preceding section.

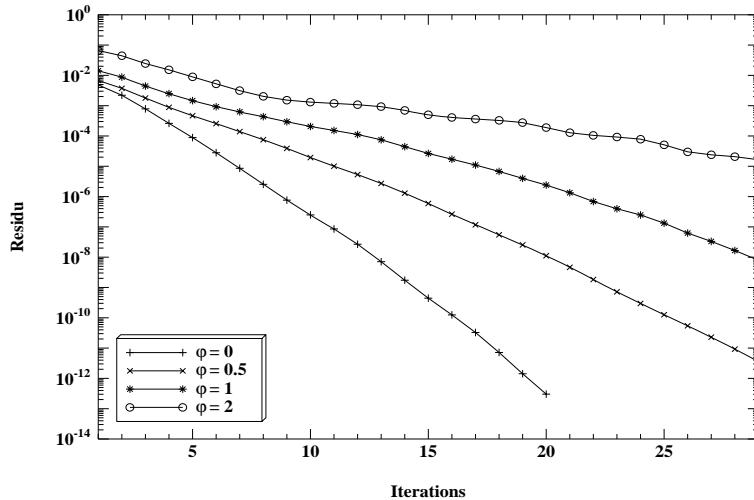
7. Numerical results

7.1. Cauchy-Riemann equations. We first present some numerical results for the Cauchy-Riemann equations (5). The computational domain is the unit square $[0, 1] \times [0, 1]$, with homogeneous boundary conditions $A_n^- U = 0$ on the boundaries $x = 0$ et $x = 1$, and periodic boundary conditions in the y direction. The subdomains consist of parallel slabs $[l_i, l_{i+1}] \times [0, 1]$ of constant width. Note that a “box” decomposition $([l_i, l_{i+1}] \times [L_j, L_{j+1}])$ is possible and would lead to similar results.

The value of δt is 10^2 , and the left hand side is $(\sin(4\pi x) \cos(4\pi y), 0)$. The meshing is 40×40 . The stopping criterion for the iterative procedure is an overall residual lower than 10^{-10} . The corresponding number of iterations is given in Table 1. This result shows that the Schwarz method is indeed convergent. The linear

TABLE 1. Schwarz method for the Cauchy-Riemann equations.

Number of Subdomains	Iterations
2	61
4	63
5	65
8	70
10	76

FIGURE 2. History of the residual, Schwarz method for the Cauchy-Riemann equations. The case $\varphi = 0$ corresponding to the absorbing boundary condition yields the best convergence.

growth with the number of subdomains is a usual feature of one-level Schwarz methods.

For practical applications, it is preferable to use the Schwarz method as a preconditioner for a Krylov subspace method.

7.2. Acceleration via GMRES. From now on, we use the Schwarz method as a preconditioner for GMRES [14]. With this approach, the number of iterations required to reach convergence for the Cauchy-Riemann system is typically of the order of 20 (see Fig. 2). This contrasts with the 60 iterations needed for the Schwarz method alone.

The main issue now becomes the condition number of the method. As explained in the preceding section, the transmission conditions have a significant impact on the behaviour of the method. Figure 2 shows the history of the residual for several transmission conditions of the type $A^{neg}u_i = A^{neg}u_j$ with two subdomains. The best behaviour is clearly obtained with $\varphi = 0$, i.e. with the first order absorbing boundary condition.

7.3. Saint-Venant's equations. We now apply the preceding ideas to the computation of smooth, non-linear, steady-state flow. The proposed problem is a two dimensional shallow water flow over a bump.

TABLE 2. Schwarz method, Shallow water problem: iterations for several decompositions.

Decomposition	Iterations
1 × 2	23
1 × 3	29
1 × 4	35
2 × 2	25
2 × 3	38
3 × 3	37

The Saint-Venant equations read:

$$\begin{aligned}\partial_t h + \partial_x(hu) + \partial_y(hv) &= 0 \\ \partial_t hu + \partial_x(hu^2 + gh^2/2) + \partial_y(huv) &= -gh\partial_x q \\ \partial_t hv + \partial_x(huv) + \partial_y(hv^2 + gh^2/2) &= -gh\partial_y q.\end{aligned}$$

Here h is the water depth, $(hu, hv)^T$ is the momentum vector, and $q(x, y)$ is the given height of the sea bottom. The equation of the free surface is therefore $h + q$. The construction of Roe's matrix for this problem is classical. The treatment of the source term follows the work of [2]: at the discrete level, we simply end up with a right hand side in the linear system.

The test case is a subsonic flow over a circular bump, with a Froude number of approximately .42. The computational domain is the unit square and a first order absorbing boundary condition is imposed at the boundary. The mesh size is 60×60 and the time step is such that $\delta t/\delta x = 10^4$. A steady-state solution is reached within 5 time steps.

The Schwarz method is used with GMRES. Table 2 gives the number of iterations required to decrease the residual by a factor of 10^{-10} for several decompositions of the computational domain. The decomposition referred to as " $i \times j$ " consists of i subdomains in the x direction and j subdomains in the y direction. For instance, 1×4 and 2×2 refer to a decomposition in 4 slices or boxes respectively.

These results show the applicability of the method to the solution of compressible flows. The growth of the iterations with the number of subdomains seems reasonable.

8. Conclusion

We have considered linear systems of first order equations. A Schwarz iterative method has been defined and the convergence of the algorithm has been studied.

Numerically, a first order finite volume discretization of the equations has been considered. A Schur complement formulation has been proposed. The influence of the number of subdomains and of the transmission condition has been investigated.

Finally, we have shown the applicability of the algorithm to a non-linear flow computation. We have used the linearly implicit version of Roe's scheme for a two-dimensional shallow water problem.

A better preconditioning might however be necessary for problems in 3-D or involving a great number of subdomains. For this purpose, a more complete numerical analysis of domain decomposition methods for first order systems is needed.

References

1. C. Bardos, D. Brézis, and H. Brézis, *Perturbations singulières et prolongements maximaux d'opérateurs positifs*, Arch. Ration. Mech. Anal. **53** (1973), 69–100.
2. A. Bermudez and E. Vazquez, *Upwind methods for hyperbolic conservation laws with source terms*, Computers & Fluids (1994), 1049–1071.
3. S. Clerc, *Etude de schémas décentrés implicites pour le calcul numérique en mécanique des fluides, résolution par décomposition de domaine*, Ph.D. thesis, Université Paris VI, 1997.
4. B. Després, *Domain decomposition and the helmholtz problem*, C.R. Acad. Sci. Paris Série I **311** (1990), 313.
5. B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp. **31** (1977), 629–651.
6. F. Gastaldi and L. Gastaldi, *Domain decomposition for the transport equation*, IMA Journal of Numer. Anal. **14** (1993), 147–165.
7. W.D. Gropp and D.E. Keyes, *Domain decomposition methods in computational fluid dynamics*, Int. J. Numer. Meth. in Fluids **14** (1992), 147–165.
8. A. Harten, *On the symmetric form of systems of conservation laws with entropy*, J.C.P. **49** (1983), 151–164.
9. L. Hemmingson, *A domain decomposition method for almost incompressible flow*, Computers & Fluids **25** (1996), 771–789.
10. P.D. Lax and R.S. Phillips, *Local boundary conditions for dissipative symmetric linear differential operators*, Comm. Pure Appl. Math. **13** (1960), 427–454.
11. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Math. Models and Methods in Appl. Sci. **5** (1995).
12. A. Quarteroni, *Domain decomposition method for systems of conservation laws: spectral collocation approximations*, SIAM J. Sci. Stat. Comput. **11** (1990), 1029–1052.
13. P.L. Roe, *Approximate riemann solvers, parameter vectors, and difference schemes*, J.C.P. **43** (1981), 357–372.
14. Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7** (1986), 856.
15. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances, vol. 1, North-Holland, 1994, pp. 121–220.
16. B. van Leer, W.-T. Lee, and P.L. Roe, *Characteristic time-stepping or local preconditioning of the Euler equations*, AIAA Paper 91-1552, AIAA, 1991.
17. Y. Wu, X.-C. Cai, and D.E. Keyes, *Additive Schwarz methods for hyperbolic equations*, this conference, 1997.

SERMA - LETR, BAT 470, COMMISSARIAT À L'ÉNERGIE ATOMIQUE, 91191 GIF-SUR-YVETTE
FRANCE

E-mail address: Sébastien.Clerc@cea.fr

Interface Conditions and Non-overlapping Domain Decomposition Methods for a Fluid–Solid Interaction Problem

Xiaobing Feng

1. Introduction

We present some non-overlapping domain decomposition methods for an inviscid fluid–solid interaction model which was proposed in [8] for modeling elastic wave propagation through fluid–filled borehole environments. Mathematically, the model is described by the coupled system of the elastic wave equations and the acoustic wave equation. First, we give a rigorous mathematical derivation of the interface conditions used in the model and to introduce some new variants of the interface conditions. The new interface conditions are mathematically equivalent to the original interface conditions, however, they can be more conveniently used to construct effective non-overlapping domain decomposition methods for the fluid–solid interaction problem. Then, we construct and analyze some parallelizable non-overlapping domain decomposition iterative methods for the fluid–solid interaction model based on the proposed new interface conditions.

The problems of wave propagation in composite media have long been subjects of both theoretical and practical studies, important applications of such problems are found in inverse scattering, elastodynamics, geosciences, oceanography. For some recent developments on modeling, mathematical and numerical analysis, and computational simulations, we refer to [2, 3, 8, 7, 10, 12] and the references therein.

The non-overlapping domain decomposition iterative methods developed in this paper are based on the idea of using the convex combinations of the interface conditions in place of the original interface conditions to pass the information between subdomains, see [9, 1, 4, 6] for the expositions and discussions on this approach for uncoupled homogeneous problems. On the other hand, for the heterogeneous fluid–solid interaction problem, it is more delicate to employ the idea because using straightforward combinations of the original interface conditions as the transmission conditions may lead to divergent iterative procedures. So the domain decomposition methods of this paper may be regarded as the generalizations

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 65F10, 73D30, 76Q05.

of the methods proposed in [1, 4, 6, 9] to the time-dependent heterogeneous problems. For more discussions on heterogeneous domain decomposition methods, we refer to [11] and the references therein.

The organization of this paper is as follows. In Section 2, the fluid-solid interaction model is introduced. In Section 3, the interface conditions, which are part of the model, are first derived using the physical arguments and then proved rigorously using vanishing shear modulus approximation. In Section 4, some parallelizable non-overlapping domain decomposition algorithms are proposed for solving the fluid-solid interaction problem. It is proved that these algorithms converge strongly in the energy spaces of the underlying fluid-solid interaction problem.

2. Description of the problem

We consider the propagation of waves in a composite medium Ω which consists of a fluid part Ω_f and a solid part Ω_s , that is, $\Omega = \Omega_f \cup \Omega_s$. Ω will be identified with a domain in \mathbb{R}^N for $N = 2, 3$, and will be taken to be of unit thickness when $N = 2$. Let $\Gamma = \partial\Omega_f \cap \partial\Omega_s$ denote the interface between two media, and let $\Gamma_f = \partial\Omega_f \setminus \Gamma$ and $\Gamma_s = \partial\Omega_s \setminus \Gamma$. Suppose that the solid is a pure elastic medium and the fluid is a pure acoustic media (inviscid fluids). Then the wave propagation is described by the following systems of partial differential equations

- (1) $\frac{1}{c^2} p_{tt} - \Delta p = g_f,$ in $\Omega_f,$
- (2) $\rho_s \mathbf{u}_{tt} - \operatorname{div}(\sigma(\mathbf{u})) = \mathbf{g}_s,$ in $\Omega_s,$
- (3) $\frac{\partial p}{\partial n_f} - \rho_f \mathbf{u}_{tt} \cdot n_s = 0,$ on $\Gamma,$
- (4) $\sigma(\mathbf{u}) n_s - p n_f = 0,$ on $\Gamma,$
- (5) $\frac{1}{c} p_t + \frac{\partial p}{\partial n_f} = 0,$ on $\Gamma_f,$
- (6) $\rho_s \mathcal{A}_s \mathbf{u}_t + \sigma(\mathbf{u}) n_s = 0,$ on $\Gamma_s,$
- (7) $p(x, 0) = p_0(x), \quad p_t(x, 0) = p_1(x),$ in $\Omega_f,$
- (8) $\mathbf{u}(x, 0) = \mathbf{u}_0(x), \quad \mathbf{u}_t(x, 0) = \mathbf{u}_1(x),$ in $\Omega_s,$

where

$$(9) \quad \sigma(\mathbf{u}) = \lambda_s \operatorname{div} \mathbf{u} I + 2\mu_s \epsilon(\mathbf{u}), \quad \epsilon(\mathbf{u}) = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T].$$

In the above description, p is the pressure function in Ω_f and \mathbf{u} is the displacement vector in Ω_s . ρ_i ($i = f, s$) denotes the density of Ω_i , n_i ($i = f, s$) denotes the unit outward normal to $\partial\Omega_i$. $\lambda_s > 0$ and $\mu_s \geq 0$ are the Lamé constants of Ω_s . Equation (9) is the constitutive relation for Ω_s . I stands for the $N \times N$ identity matrix. The boundary conditions in (5) and (6) are the first order absorbing boundary conditions for acoustic and the elastic waves, respectively. These boundary conditions are transparent to waves arriving normally at the boundary (cf. [5]). Finally, equations (3) and (4) are the interface conditions which describe the interaction between the fluid and the solid.

A derivation of the above model can be found in [8], where a detailed mathematical analysis concerning the existence, uniqueness and regularity of the solutions was also presented. The finite element approximations of the model were studied in [7], both semi-discrete and fully-discrete finite element methods were proposed

and optimal order error estimates were obtained in both cases for the fluid-solid interaction model.

3. Interface conditions

The purpose of this section is to present a rigorous mathematical justification and interpretation for the interface conditions (3) and (4), which describe the interaction between the fluid and the solid. These interface conditions were originally derived in [8] using the heuristic physical arguments. For the sake of completeness, we start this section with a brief review of the heuristic derivation.

3.1. Derivation of interface conditions by physical arguments. Since an acoustic media can be regarded as an elastic media with zero shear modulus, the motion of each part of the composite media is described by a system of the elastic wave equations

$$(10) \quad \rho_i(\mathbf{u}_i)_{tt} = \operatorname{div} \sigma(\mathbf{u}_i) + \mathbf{g}_i, \quad \text{in } \Omega_i,$$

$$(11) \quad \sigma(\mathbf{u}_i) = \lambda_i \operatorname{div} \mathbf{u}_i I + 2\mu_i \varepsilon(\mathbf{u}_i),$$

$$(12) \quad \varepsilon(\mathbf{u}_i) = \frac{1}{2}[\nabla \mathbf{u}_i + (\nabla \mathbf{u}_i)^T],$$

where $i = f, s$. \mathbf{u}_i denotes the displacement vector in Ω_i , and $\mu_f = 0$. Notice that the displacement is used as the primitive variable in both fluid and solid region.

Physically, as a system, the following two conditions must be satisfied on the interface between the fluid and the solid (cf. [3] and references therein).

- No relative movements occur in the normal direction of the interface.
- The stress must be continuous across the interface.

Mathematically, these conditions can be formulated as

$$(13) \quad \mathbf{u}_s \cdot \mathbf{n}_s + \mathbf{u}_f \cdot \mathbf{n}_f = 0, \quad \text{on } \Gamma,$$

$$(14) \quad \sigma(\mathbf{u}_s)\mathbf{n}_s + \sigma(\mathbf{u}_f)\mathbf{n}_f = 0, \quad \text{on } \Gamma.$$

In practice, it is more convenient to use the pressure field in the acoustic medium, so it is necessary to rewrite the interface conditions (13) and (14) using the pressure–displacement formulation. To this end, we introduce the pressure of the fluid, $p = -\lambda_f \operatorname{div} \mathbf{u}_f$. Since $\mu_f = 0$, (14) can be rewritten as

$$\sigma(\mathbf{u}_s)\mathbf{n}_s = -\sigma(\mathbf{u}_f)\mathbf{n}_f = pn_f, \quad \text{on } \Gamma.$$

which gives the interface condition (4). To convert (13), differentiating it twice with respect to t and using (10) we get

$$\rho_f(\mathbf{u}_s \cdot \mathbf{n}_s)_{tt} = -\rho_f(\mathbf{u}_f \cdot \mathbf{n}_f)_{tt} = -\operatorname{div}(\sigma(\mathbf{u}_f)) \cdot \mathbf{n}_f = \frac{\partial p}{\partial n_f}, \quad \text{on } \Gamma,$$

so we get (13). Here we have used the fact that the source \mathbf{g}_f vanishes on the interface Γ .

Finally, to get the full model (1)–(9) we also need to transform the interior equation in the fluid region from the displacement formulation into the pressure formulation. For a detailed derivation of this transformation, we refer to [8].

3.2. Validation of interface conditions by vanishing shear modulus approximation. The goal of this subsection is to show the interface conditions (13) and (14) are proper conditions to describe the interaction between the fluid region and solid region. In the same time, our derivation also reveals that in what sense the conditions (13) and (14) hold. To do this, we regularize the constitutive equation of the fluid by introducing a small (artificial) shear modulus $\mu_f = \delta > 0$, and then to look for the limiting model of the regularized problem as δ goes to zero.

The regularized constitutive equation of the fluid reads as

$$(15) \quad \sigma^\delta(\mathbf{u}_f) = \lambda_f(\operatorname{div} \mathbf{u}_f)I + 2\delta\varepsilon(\mathbf{u}_f).$$

Let $(\mathbf{u}_f^\delta, \mathbf{u}_s^\delta)$ be the solution of the regularized problem (10)–(12) and (15) with prescribed boundary conditions (say, first order absorbing boundary conditions) and initial conditions. It is well-known that the interface conditions for second order elliptic problems are the continuity of the function value and the continuity of the normal flux across the interface (cf. [9]). For the regularized problem (10)–(12) and (15), this means that

$$(16) \quad \mathbf{u}_f^\delta = \mathbf{u}_s^\delta, \quad \text{on } \Gamma,$$

$$(17) \quad \sigma^\delta(\mathbf{u}_f^\delta)n_f = -\sigma(\mathbf{u}_s^\delta)n_s, \quad \text{on } \Gamma.$$

The main result of this section is the following convergence theorem.

THEOREM 1. *There exist $\mathbf{u}_f \in H(\operatorname{div}, \Omega_f)$ and $\mathbf{u}_s \in H^1(\Omega_s)$ such that*

- (i) \mathbf{u}_f^δ converges to \mathbf{u}_f weakly in $H(\operatorname{div}, \Omega_f)$.
- (ii) \mathbf{u}_s^δ converges to \mathbf{u}_s weakly in $H^1(\Omega_s)$.
- (iii) $(\mathbf{u}_f, \mathbf{u}_s)$ satisfies the interface conditions (13) and (14) in $(H_{00}^{\frac{1}{2}}(\Gamma))'$.

PROOF. Due to the page limitation, we only sketch the idea of the proof. To see a detailed proof of similar type, we refer to [8, 11].

The idea of the proof is to use the energy method. To apply the energy method, the key step is to get the uniform (in δ) estimates for the solution $(\mathbf{u}_f^\delta, \mathbf{u}_s^\delta)$. This can be done by testing the interior equations of the fluid and solid against $(\mathbf{u}_f^\delta)_t$ and $(\mathbf{u}_s^\delta)_t$, respectively. Finally, the proof is completed by using a compactness argument and taking limit in (10)–(12) and (15)–(17) as δ goes to zero. \square

4. Non-overlapping domain decomposition methods

Because of the existence of the physical interface, it is very nature to use non-overlapping domain decomposition method to solve the fluid–solid interaction problem. In fact, Non-overlapping domain decomposition methods have been effectively used to solve several coupled boundary value problems from scientific applications, see [11] and the references therein.

In this section we first propose a family of new interface conditions which are equivalent to the original interface conditions (3) and (4). From a mathematical point of view, this is the key step towards developing non-overlapping domain decomposition methods for the problem. Based on these new interface conditions, we introduce two types of parallelizable non-overlapping domain decomposition iterative algorithms for solving the system (1)–(9) and establish the usefulness of

these algorithms by proving their strong convergence in the energy spaces of the underlying fluid–solid interaction problem.

Due to the page limitation, the algorithms and the analyses are only given at the differential level in this paper. Following the ideas of [1, 4, 6], it is not very hard rather technical and tedious to construct and analyze the finite element discrete analogues of the differential domain decomposition algorithms. Another point which is worth mentioning is that the domain decomposition algorithms of this paper can be used for solving the discrete systems of (1)–(9) which arise from using other discretization methods such as finite difference and spectral methods, as well as hybrid methods of using different discretization methods in different media (subdomains).

4.1. Algorithms. Recall the interface conditions on the fluid–solid contact surface are

$$(18) \quad \frac{\partial p}{\partial n_f} = \rho_f \mathbf{u}_{tt} \cdot n_s, \quad p n_f = \sigma(\mathbf{u}) n_s, \quad \text{on } \Gamma.$$

Rewrite the second equation in (18) as

$$(19) \quad -p_t = \sigma(\mathbf{u}_t) n_s \cdot n_s, \quad 0 = \sigma(\mathbf{u}_t) n_s \cdot \tau_s, \quad \text{on } \Gamma,$$

where τ_s denotes the unit tangential vector on $\partial\Omega_s$. The equivalence of (18)₂ and (19) holds if the initial conditions satisfy some compatibility conditions (cf. [8]).

LEMMA 2. *The interface conditions in (18) are equivalent to*

$$(20) \quad \frac{\partial p}{\partial n_f} + \alpha p_t = \rho_f \mathbf{u}_{tt} \cdot n_s - \alpha \sigma(\mathbf{u}_t) n_s \cdot n_s, \quad \text{on } \Gamma,$$

$$(21) \quad \rho_f \mathbf{u}_{tt} + \beta \sigma(\mathbf{u}_t) n_s = \frac{\partial p}{\partial n_f} n_s - \beta p_t n_s, \quad \text{on } \Gamma,$$

$$(22) \quad \sigma(\mathbf{u}_t) n_s \tau_s = 0, \quad \text{on } \Gamma,$$

for any pair of constants α and β such that $\alpha + \beta \neq 0$.

Based on the above new form of the interface conditions we propose the following two types of iterative algorithms. The first one resembles to Jacobi type iteration and the other resembles to Gauss–Seidel type iteration.

Algorithm 1

Step 1 $\forall p^0 \in P_f, \forall \mathbf{u}^0 \in \mathbf{V}_s$.

Step 2 Generate $\{(p^n, \mathbf{u}^n)\}_{n \geq 1}$ iteratively by solving

$$(23) \quad \frac{1}{c^2} p_{tt}^n - \Delta p^n = g_f, \quad \text{in } \Omega_f,$$

$$(24) \quad \frac{1}{c} p_t^n + \frac{\partial p^n}{\partial n_f} = 0, \quad \text{on } \Gamma_f,$$

$$(25) \quad \frac{\partial p^n}{\partial n_f} + \alpha p_t^n = \rho_f \mathbf{u}_{tt}^{n-1} \cdot n_s - \alpha \sigma(\mathbf{u}_t^{n-1}) n_s \cdot n_s, \quad \text{on } \Gamma;$$

$$(26) \quad \rho_s \mathbf{u}_{tt}^n - \operatorname{div} \sigma(\mathbf{u}^n) = \mathbf{g}_s, \quad \text{in } \Omega_s,$$

$$(27) \quad \rho_s \mathcal{A}_s \mathbf{u}_t^n + \sigma(\mathbf{u}^n) n_s = 0, \quad \text{on } \Gamma_s,$$

$$(28) \quad \rho_f \mathbf{u}_{tt}^n + \beta \sigma(\mathbf{u}_t^n) n_s = \frac{\partial p^{n-1}}{\partial n_f} n_s - \beta p_t^{n-1} n_s, \quad \text{on } \Gamma,$$

$$(29) \quad \sigma(\mathbf{u}_t^n) n_s \cdot \tau_s = 0, \quad \text{on } \Gamma.$$

Algorithm 2Step 1 $\forall \mathbf{u}^0 \in \mathbf{V}_s$.Step 2 Generate $\{p^n\}_{n \geq 0}$ and $\{\mathbf{u}^n\}_{n \geq 1}$ iteratively by solving

(30)
$$\frac{1}{c^2} p_{tt}^n - \Delta p^n = g_f, \quad \text{in } \Omega_f,$$

(31)
$$\frac{1}{c} p_t^n + \frac{\partial p^n}{\partial n_f} = 0, \quad \text{on } \Gamma_f,$$

(32)
$$\frac{\partial p^n}{\partial n_f} + \alpha p_t^n = \rho_f \mathbf{u}_{tt}^n \cdot \mathbf{n}_s - \alpha \sigma(\mathbf{u}_t^n) \mathbf{n}_s \cdot \mathbf{n}_s, \quad \text{on } \Gamma;$$

(33)
$$\rho_s \mathbf{u}_{tt}^{n+1} - \operatorname{div} \sigma(\mathbf{u}^{n+1}) = \mathbf{g}_s, \quad \text{in } \Omega_s,$$

(34)
$$\rho_s \mathcal{A}_s \mathbf{u}_t^{n+1} + \sigma(\mathbf{u}^{n+1}) \mathbf{n}_s = 0, \quad \text{on } \Gamma_s,$$

(35)
$$\rho_f \mathbf{u}_{tt}^{n+1} + \beta \sigma(\mathbf{u}_t^{n+1}) \mathbf{n}_s = \frac{\partial p^n}{\partial n_f} \mathbf{n}_s - \beta p_t^n \mathbf{n}_s, \quad \text{on } \Gamma,$$

(36)
$$\sigma(\mathbf{u}_t^{n+1}) \mathbf{n}_s \cdot \tau_s = 0, \quad \text{on } \Gamma.$$

REMARK 3. Appropriate initial conditions must be provided in the above algorithms. We omit these conditions for notation brevity.

4.2. Convergence Analysis. In this subsection we shall establish the utility of Algorithms 1 and 2 by proving their convergence. Because the convergence proof for Algorithm 2 is almost same as the proof of Algorithm 1, we only give a proof for Algorithm 1 in the following.

Introduce the error functions at the n th iteration

$$r^n = p - p^n, \quad \mathbf{e}^n = \mathbf{u} - \mathbf{u}^n.$$

It is easy to check that (r^n, \mathbf{e}^n) satisfies the error equations

(37)
$$\frac{1}{c^2} r_{tt}^n - \Delta r^n = 0, \quad \text{in } \Omega_f,$$

(38)
$$\frac{1}{c} r_t^n + \frac{\partial r^n}{\partial n_f} = 0, \quad \text{on } \Gamma_f,$$

(39)
$$\frac{\partial r^n}{\partial n_f} + \alpha r_t^n = \rho_f \mathbf{e}_{tt}^{n-1} \cdot \mathbf{n}_s - \alpha \sigma(\mathbf{e}_t^{n-1}) \mathbf{n}_s \cdot \mathbf{n}_s, \quad \text{on } \Gamma;$$

(40)
$$\rho_s \mathbf{e}_{tt}^n - \operatorname{div} \sigma(\mathbf{e}^n) = 0, \quad \text{in } \Omega_s,$$

(41)
$$\rho_s \mathcal{A}_s \mathbf{e}_t^n + \sigma(\mathbf{e}^n) \mathbf{n}_s = 0, \quad \text{on } \Gamma_s,$$

(42)
$$\rho_f \mathbf{e}_{tt}^n + \beta \sigma(\mathbf{e}_t^n) \mathbf{n}_s = \frac{\partial r^{n-1}}{\partial n_f} \mathbf{n}_s - \beta r_t^{n-1} \mathbf{n}_s, \quad \text{on } \Gamma,$$

(43)
$$\sigma(\mathbf{e}_t^n) \mathbf{n}_s \cdot \tau_s = 0, \quad \text{on } \Gamma.$$

Define the “pseudo–energy”

$$E_n = E(\{r^n, \mathbf{e}^n\}) = \left\| \frac{\partial r^n}{\partial n_f} + \alpha r_t^n \right\|_{L^2(\lg)}^2 + \|\rho_f \mathbf{e}_{tt}^n + \beta \sigma(\mathbf{e}^n) \mathbf{n}_s\|_{L^2(\lg)}^2.$$

LEMMA 4. *There holds the following inequality*

(44)
$$E_{n+1}(\tau) \leq E_n(\tau) - R_n(\tau),$$

where

$$R_n(\tau) = 4 \int_0^\tau \int_\Gamma \left[\alpha \frac{\partial r^n}{\partial n_f} r_t^n + \beta \sigma(\mathbf{e}_t^n) \mathbf{n}_s \cdot \mathbf{e}_{tt}^n \right] dx dt.$$

To make the estimate (44) be useful, we need to find a lower bound for $R_n(\tau)$. Testing (37) against r_t^n to get

$$\frac{1}{2} \frac{d}{dt} \left\| \frac{1}{c} r_t^n \right\|_{0,\Omega_f}^2 + \frac{1}{2} \frac{d}{dt} \|\nabla r^n\|_{0,\Omega_f}^2 + \left| \frac{1}{\sqrt{c}} r_t^n \right|_{0,\Gamma_f}^2 = \int_{\Gamma} \frac{\partial r^n}{\partial n_f} r_t^n dx,$$

which implies that

$$(45) \quad \int_0^\tau \int_{\Gamma} \frac{\partial r^n}{\partial n_f} r_t^n dx dt = \frac{1}{2} \left\| \frac{1}{c} r_t^n(\tau) \right\|_{0,\Omega_f}^2 + \frac{1}{2} \|\nabla r^n(\tau)\|_{0,\Omega_f}^2 + \left\| \frac{1}{\sqrt{c}} r_t^n \right\|_{L^2(L^2(\Gamma_f))}^2.$$

Here we have implicitly assumed that $r^n(0) = r_t^n(0) = 0$.

Differentiating (40) with respect to t and testing it against \mathbf{e}_{tt}^n give us

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \|\sqrt{\rho_s} \mathbf{e}_{tt}^n\|_{0,\Omega_s}^2 &+ \frac{d}{dt} \|\sqrt{\mu_s} \varepsilon(\mathbf{e}_t^n)\|_{0,\Omega_s}^2 + \frac{1}{2} \frac{d}{dt} \left\| \sqrt{\lambda_s} \operatorname{div}(\mathbf{e}_t^n) \right\|_{0,\Omega_s}^2 \\ &+ c_0 |\sqrt{\rho_s} \mathbf{e}_t^n|_{0,\Gamma_s}^2 \leq \int_{\Gamma} \sigma(\mathbf{e}_t^n) \cdot n_s \mathbf{e}_{tt}^n dx, \end{aligned}$$

which implies that

$$(46) \quad \begin{aligned} \int_0^\tau \int_{\Gamma} \rho_s \sigma(\mathbf{e}_t^n) n_s \cdot \mathbf{e}_{tt}^n dx dt &\geq \frac{1}{2} \|\sqrt{\rho_s} \mathbf{e}_{tt}^n(\tau)\|_{0,\Omega_s}^2 + \|\sqrt{\mu_s} \varepsilon(\mathbf{e}_t^n(\tau))\|_{0,\Omega_s}^2 \\ &+ \frac{1}{2} \left\| \sqrt{\lambda_s} \operatorname{div}(\mathbf{e}_t^n(\tau)) \right\|_{0,\Omega_s}^2 + c_0 \|\sqrt{\rho_s} \mathbf{e}_t^n\|_{L^2(L^2(\Gamma_s))}^2 \\ &- \frac{1}{2} \|\sqrt{\rho_s} \mathbf{e}_{tt}^n(0)\|_{0,\Omega_s}^2. \end{aligned}$$

Since $\mathbf{e}^n(0) = \mathbf{e}_t^n(0) = 0$, it follows from (37) that

$$\|\sqrt{\rho_s} \mathbf{e}_{tt}^n(0)\|_{0,\Omega_s} = \left\| \frac{1}{\sqrt{\rho_s}} \operatorname{div}(\mathbf{e}^n(0)) \right\|_{0,\Omega_s} = 0.$$

Combining (45) and (46) we get the following lemma.

LEMMA 5. $R_n(\tau)$ satisfies the following inequality

$$\begin{aligned} R_n(\tau) &\geq 2\alpha \left[\left\| \frac{1}{c} r_t^n(\tau) \right\|_{0,\Omega_f}^2 + \|\nabla r^n(\tau)\|_{0,\Omega_f}^2 + 2 \left\| \frac{1}{\sqrt{c}} r_t^n \right\|_{L^2(L^2(\Gamma_f))}^2 \right] \\ &+ 2\beta \left[\|\sqrt{\rho_s} \mathbf{e}_{tt}^n(\tau)\|_{0,\Omega_s}^2 + \|\sqrt{\mu_s} \varepsilon(\mathbf{e}_t^n(\tau))\|_{0,\Omega_s}^2 \right. \\ &\quad \left. + \|\sqrt{\lambda_s} \operatorname{div}(\mathbf{e}_t^n(\tau))\|_{0,\Omega_s}^2 + 2c_0 \|\sqrt{\rho_s} \mathbf{e}_t^n\|_{L^2(L^2(\Gamma_s))}^2 \right]. \end{aligned}$$

Finally, from Lemma 4 and 5 we get the following convergence theorem.

THEOREM 6. Let $\{(p^k, \mathbf{u}^k)\}$ be generated by Algorithm 1 or Algorithm 3. For $\alpha > 0$ and $\beta > 0$, we have

- (1) $p^k \rightarrow p$ strongly in $L^\infty(H^1(\Omega_f)) \cap W^{1,\infty}(L^2(\Omega_f))$,
- (2) $\mathbf{u}^k \rightarrow \mathbf{u}$ strongly in $W^{1,\infty}(\mathbf{H}^1(\Omega_s)) \cap W^{2,\infty}(\mathbf{L}^2(\Omega_s))$.

REMARK 7. If we choose $\alpha = 0$, $\beta = \infty$, Algorithms 1 and 2 become $N-N$ alternating type algorithms. In addition, at the end of each $N-N$ iteration one can add the following relaxation step to speed up the convergence

$$\begin{aligned} p^n &:= \mu p^n + (1 - \mu) p^{n-1}, \\ \mathbf{u}_{tt}^n &:= \mu \mathbf{u}_{tt}^n + (1 - \mu) \mathbf{u}_{tt}^{n-1}, \end{aligned}$$

where μ is any constant satisfying $0 < \mu < 1$.

5. Conclusions

In this paper we have presented a mathematically rigorous derivation of the interface conditions for the *inviscid* fluid–solid interaction model proposed in [8], and developed two families of non-overlapping domain decomposition iterative methods for solving the governing partial differential equations. Our analysis demonstrated that it is crucial and delicate to choose the *right* transmission conditions for constructing domain decomposition methods for the problem, since trivial use of the physical interface conditions as the transmission conditions may result in slowly convergent even divergent iterative methods. Finally, we believe that the ideas and methods presented in this paper can be extended to other heterogeneous problems, in particular, the *viscid* fluid–solid interaction problems in which the Navier–Stokes equations should be used in the fluid medium. This work is currently in progress and will be reported in the near future.

References

1. 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., Modélisation Math. Anal. Numér. **31** (1997), 1–25.
2. J. Boujot, *Mathematical formulation of fluid–structure interaction problems*, Math., Modeling and Numer. Anal. **21** (1987), 239–260.
3. L. Demkowicz, J. T. Oden, M. Ainsworth, and P. Geng, *Solution of elastic scattering problems in linear acoustic using h - p boundary element methods*, J. Comp. Appl. Math. **36** (1991), 29–63.
4. J. Douglas, Jr., P. J. S. Paes Leme, J. E. Roberts, and J. Wang, *A parallel iterative procedure applicable to the approximate solution of second order partial differential equations by mixed finite element methods*, Numer. Math. **65** (1993), 95–108.
5. B. Engquist and A. Majda, *Radiation boundary conditions for acoustic and elastic wave calculations*, Comm. Pure Appl. Math. **32** (1979), 313–357.
6. X. Feng, *Analysis of a domain decomposition method for the nearly elastic wave equations based on mixed finite element methods*, IMA J. Numer. Anal. **14** (1997), 1–22.
7. X. Feng, P. Lee, and Y. Wei, *Finite element methods and domain decomposition algorithms for a fluid–solid interaction problem*, Tech. Report 1496, Institute of Mathematics and Applications, University of Minnesota, 1997.
8. ———, *Formulation and mathematical analysis of a fluid–solid interaction problem*, Tech. Report 1495, Institute of Mathematics and Applications, University of Minnesota, 1997, (to appear in Mat. Apl. Comput.).
9. P. L. Lions, *On the Schwartz alternating method III*, Proceedings of Third International Symposium on Domain Decomposition Method for Partial Differential Equations, SIAM, Philadelphia, 1990, pp. 202–223.
10. Ch. Makridakis, F. Ihlenburg, and I. Babuška, *Analysis and finite element methods for a fluid–structure interaction problem in one dimension*, Tech. Report BN-1183, IPST, University of Maryland at College Park, 1995.
11. A. Quarteroni, F. Pasquarelli, and A. Valli, *Heterogeneous domain decomposition: principles, algorithms, applications*, Proceedings of Fifth International Symposium on Domain Decomposition Method for Partial Differential Equations, SIAM, Philadelphia, 1992, pp. 129–150.
12. J. E. Santos, J. Douglas, Jr., and A. P. Calderón, *Finite element methods for a composite model in elastodynamics*, SIAM J. Numer. Anal. **25** (1988), 513–523.

Overlapping Schwarz Waveform Relaxation for Parabolic Problems

Martin J. Gander

1. Introduction

We analyze a new domain decomposition algorithm to solve parabolic partial differential equations. Two classical approaches can be found in the literature:

1. Discretizing time and applying domain decomposition to the obtained sequence of elliptic problems, like in [9, 11, 1] and references therein.
2. Discretizing space and applying waveform relaxation to the large system of ordinary differential equations, like in [10, 8, 7] and references therein.

In contrary to the classical approaches, we formulate a parallel solution algorithm without any discretization. We decompose the domain into overlapping subdomains in space and we consider the parabolic problem on each subdomain over a given time interval. We solve iteratively parabolic problems on subdomains, exchanging boundary information at the interfaces of subdomains. So for a parabolic problem $\frac{\partial u}{\partial t} = \mathcal{L}(u, x, t)$ in the domain Ω with given initial and boundary conditions the algorithm for two overlapping subdomains Ω_0 and Ω_1 would read for $j = 0, 1$,

$$\begin{aligned}\frac{\partial u_j^{k+1}}{\partial t} &= \mathcal{L}(u_j^{k+1}, x, t), \quad x, t \in \Omega_j \\ u_j^{k+1}(x, t) &= \begin{cases} u_{1-j}^k(x, t), & x, t \in \Gamma_j := \partial\Omega_j \cap \Omega_{1-j} \\ \text{given bc,} & x, t \in \partial\Omega_j - \Gamma_j \end{cases}\end{aligned}$$

This algorithm is like a classical overlapping additive Schwarz, but on subdomains, a time dependent problem is solved, like in waveform relaxation. Thus the name overlapping Schwarz waveform relaxation.

This algorithm has been considered before in [4], where linear convergence on unbounded time intervals was proved for the one dimensional heat equation, and in [6] where superlinear convergence for bounded time intervals was shown for a constant coefficient convection diffusion equation.

We study here three model problems which show that the results in [4] and [6] hold for more general parabolic equations. All the convergence results are in L^∞ with the norm $\|u\|_\infty := \sup_{x,t} |u(x, t)|$.

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 76R99.

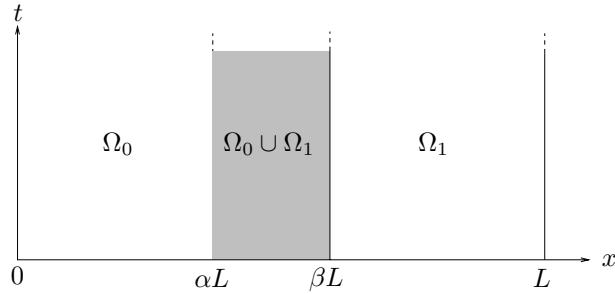


FIGURE 1. Decomposition into two overlapping subdomains

2. Reaction Diffusion Equation

We study the overlapping Schwarz waveform relaxation algorithm for the one dimensional reaction diffusion equation with variable diffusion coefficient

$$\frac{\partial u}{\partial t} = c^2(x, t) \frac{\partial^2 u}{\partial x^2} + f(u) \quad 0 < x < L, \quad 0 < t < T$$

with appropriate initial and boundary conditions. We consider only the two subdomain problem given in Figure 1 with nonempty overlap, $0 < \alpha < \beta < 1$. The generalization to N subdomains can be found in [2].

2.1. Theoretical Results. Define for $0 < x < L$ and $0 < t < T$

$$\hat{c}^2 := \sup_{x,t} c^2(x, t), \quad \hat{a} := \sup_{x,t,\xi \in \mathbb{R}} \frac{f'(\xi)}{c^2(x, t)}$$

which are assumed to be finite.

THEOREM 1 (Linear convergence for $t \in [0, \infty)$). *If $-\infty < \hat{a} < (\frac{\pi}{L})^2$ then the overlapping Schwarz waveform relaxation algorithm converges linearly on unbounded time intervals,*

$$\max_j \|u - u_j^{2k+1}\|_\infty \leq \gamma^k C \max_j \|u - u_j^0\|_\infty$$

with convergence factor

$$\gamma = \frac{\sin(\sqrt{\hat{a}}(1-\beta)L) \cdot \sin(\sqrt{\hat{a}}\alpha L)}{\sin(\sqrt{\hat{a}}(1-\alpha)L) \cdot \sin(\sqrt{\hat{a}}\beta L)} < 1$$

and C a constant depending on the parameters of the problem.

PROOF. The proof can be found in [3]. □

Note that the convergence factor γ tends to 1 and convergence is lost as the overlap goes to zero or \hat{a} goes to $(\pi/L)^2$. On the other hand for \hat{a} negative, the sine functions in γ become hyperbolic sine functions and $\gamma \rightarrow 0$ as $\hat{a} \rightarrow -\infty$.

THEOREM 2 (Superlinear convergence for $t \in [0, T]$). *The overlapping Schwarz waveform relaxation algorithm converges superlinearly on bounded time intervals,*

$$\max_j \|u - u_j^{2k+1}\|_\infty \leq \max(1, e^{2\hat{c}^2 \hat{a} T}) \operatorname{erfc}\left(\frac{k(\beta - \alpha)L}{\sqrt{\hat{c}^2 T}}\right) \max_j \|u - u_j^0\|_\infty.$$

PROOF. The proof can be found in [3]. □

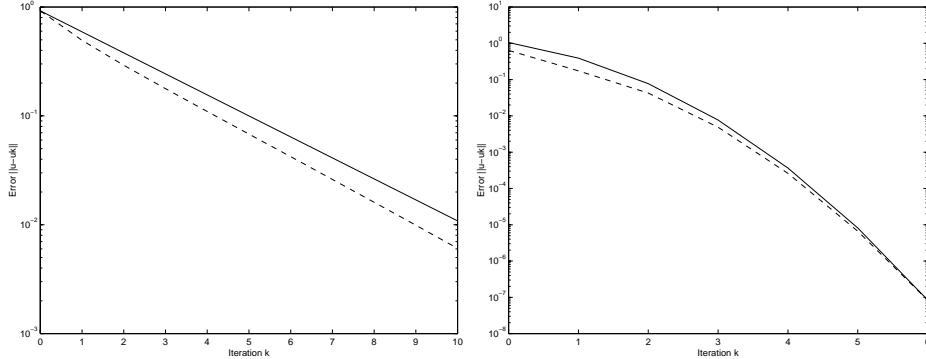


FIGURE 2. Linear convergence on long time intervals on the left and superlinear convergence on short time intervals on the right. Dashed the numerical results and solid the derived upper bounds for the convergence rate

Note that convergence is lost again when the overlap goes to zero ($\alpha = \beta$). But in contrary to the linear result, the superlinear convergence rate does not depend on \hat{u} , only the constant in front does. Instead there is a dependence on the diffusion coefficient \hat{c}^2 and the length of the time interval: the smaller the product $\hat{c}^2 T$, the faster the convergence.

2.2. Numerical Results.

We consider the model problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + 5(u - u^3), \quad 0 < x < 1, \quad 0 < t < T$$

with initial condition $u(x, 0) = x^2$ and boundary conditions $u(0, t) = 0$ and $u(1, t) = e^{-t}$. We chose 20% overlap ($\alpha = 0.4$ and $\beta = 0.6$). On the left of Figure 2 we show the algorithm in the linear convergence regime for a long time interval $T = 3$. Since this is still a bounded time interval, the algorithm would start to exhibit superlinear convergence if the iteration was continued. On the right of Figure 2 we chose a short time interval $T = 0.1$ to see the algorithm directly in the superlinear convergence regime. We used a centered second order finite difference in space with $\Delta x = 0.01$ and backward Euler in time with 300 time steps and the error displayed is the difference between the numerical solution on the whole domain and the iterates on the subdomains.

3. Convection Diffusion Equation

We consider a convection diffusion equation with variable coefficients in one dimension

$$\frac{\partial u}{\partial t} = c^2(x, t) \frac{\partial^2 u}{\partial x^2} + b(x, t) \frac{\partial u}{\partial x} + f(x, t) \quad 0 < x < L, \quad 0 < t < T$$

with appropriate initial and boundary conditions.

3.1. Theoretical Results.

Define for $0 < x < L$ and $0 < t < T$

$$\hat{c}^2 := \sup_{x,t} c^2(x, t), \quad \hat{b} := \sup_{x,t} \frac{b(x, t)}{c^2(x, t)}, \quad \check{b} := \inf_{x,t} \frac{b(x, t)}{c^2(x, t)}$$

which are assumed to be finite.

THEOREM 3 (Linear convergence for $t \in [0, \infty)$). *The overlapping Schwarz waveform relaxation algorithm converges linearly on unbounded time intervals,*

$$\max_j \|u - u_j^{2k+1}\|_\infty \leq \gamma^k \max_j \|u - u_j^0\|_\infty$$

with convergence factor

$$\gamma = \frac{1 - e^{-\tilde{b}(\beta L - L)}}{1 - e^{-\tilde{b}(\alpha L - L)}} \cdot \frac{1 - e^{-\hat{b}\alpha L}}{1 - e^{-\hat{b}\beta L}}.$$

PROOF. The proof can be found in [2]. □

As in the reaction diffusion case convergence is lost as the overlap goes to zero. But a large ratio of convection to diffusion helps the algorithm by decreasing γ .

THEOREM 4 (Superlinear convergence for $t \in [0, T]$). *The overlapping Schwarz waveform relaxation algorithm converges superlinearly on bounded time intervals,*

$$\max_j \|u - u_j^{2k+1}\|_\infty \leq e^{(\beta - \alpha)L(\hat{b} - \tilde{b})k/2} \cdot \operatorname{erfc}\left(\frac{k(\beta - \alpha)L}{\sqrt{\tilde{c}^2 T}}\right) \max_j \|u - u_j^0\|_\infty$$

PROOF. The proof can be found in [2]. □

Note that the principal convergence rate is the same as in the reaction diffusion case, but there is a lower order dependence on the convection term in the bound. The dependence is lower order, because $\operatorname{erfc}(x) \leq e^{-x^2}$. If the coefficients are constant, the dependence on the convection term disappears completely.

3.2. Numerical Results.

We consider the model problem

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + b \frac{\partial u}{\partial x} + 5e^{-(t-2)^2 - (x-\frac{1}{4})^2} \quad 0 < x < 1, \quad 0 < t < T$$

with the same initial and boundary conditions as before. We chose again an overlap of 20% and different convection terms, $b \in \{0, 2.5, 5\}$. Figure 3 shows on the left the algorithm in the linear convergence regime for $T = 3$ and the three different values of b and on the right the algorithm in the superlinear convergence regime for $T = 0.1$ and $b = 2.5$. We used again a finite difference scheme with $\Delta x = 0.01$ and backward Euler in time with 300 time steps.

4. Heat Equation in n -Dimensions

Consider the heat equation in n dimensions

$$\frac{\partial u}{\partial t} = \Delta u + f(\mathbf{x}, t) \quad \mathbf{x} \in \Omega, \quad 0 < t < T$$

with appropriate initial and boundary conditions. We assume that Ω is a smoothly bounded domain in \mathbb{R}^n and that we have an overlapping decomposition of the domain into a finite number of subdomains [5].

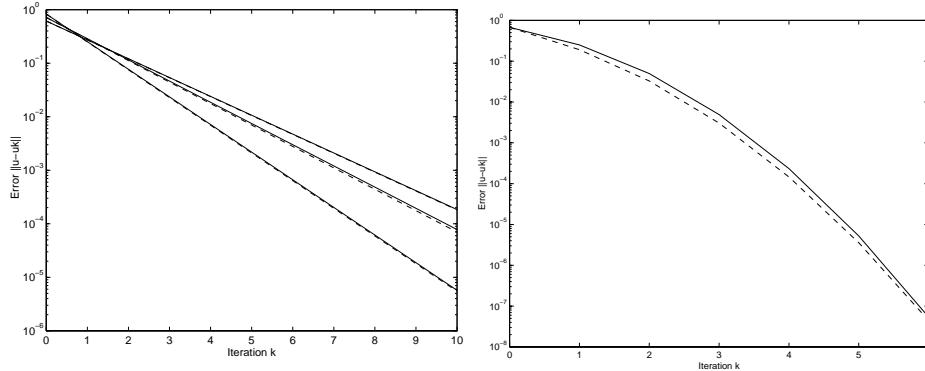


FIGURE 3. Linear convergence on the left and superlinear convergence on the right, dashed the numerical experiment and solid the theoretical bound on the convergence rate.

4.1. Theoretical Results.

THEOREM 5 (Linear convergence for $t \in [0, \infty)$). *The overlapping Schwarz waveform relaxation algorithm converges linearly on unbounded time intervals*

$$\max_j \|u - u_j^{k+m+1}\|_\infty \leq \gamma \max_j \|u - u_j^k\|_\infty$$

where $\gamma < 1$, dependent on the geometry but independent of k . Note that m depends on the number of subdomains.

PROOF. The proof can be found in [5]. □

THEOREM 6 (Superlinear convergence for $t \in [0, T]$). *The overlapping Schwarz waveform relaxation algorithm converges superlinearly on bounded time intervals*

$$\max_j \|u - u_j^k\|_\infty \leq (2n)^k \operatorname{erfc}(\frac{k\delta}{2\sqrt{nT}}) \max_j \|u - u_j^0\|_\infty$$

where δ is a parameter related to the size of the overlap.

PROOF. The proof can be found in [5]. □

Note that the superlinear convergence rate does not depend on the number of subdomains.

4.2. Numerical Results. We used the homogeneous heat equation in two dimensions with homogeneous initial and boundary conditions on a unit square with the decomposition given in Figure 4 for our numerical experiments. Figure 5 shows for two different sizes of the overlap, $(\alpha, \beta) \in \{(0.3, 0.7), (0.4, 0.6)\}$, on the left the linear convergence behavior of the algorithm for a long time interval, $T = 3$ and on the right the superlinear convergence behavior for a short time interval, $T = 0.1$. We used a centered second order finite difference scheme in space with $\Delta x = 1/30$ and backward Euler in time with 100 time steps.

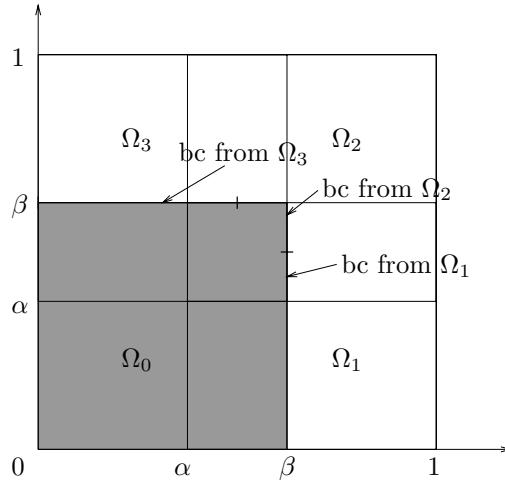


FIGURE 4. Decomposition of the domain for the heat equation model problem

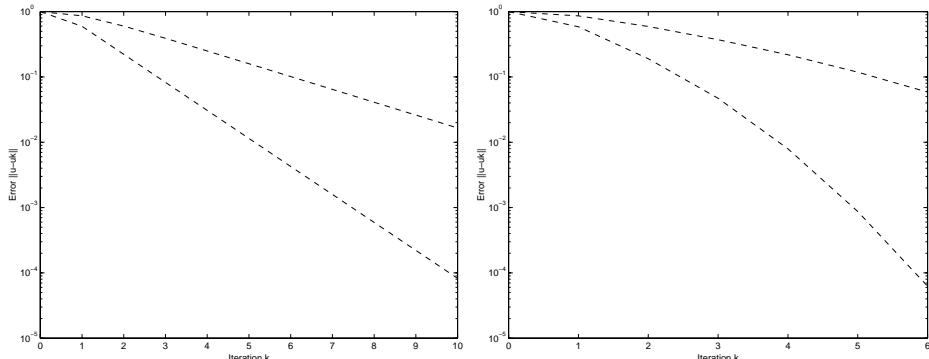


FIGURE 5. Linear convergence on the left and superlinear convergence on the right for the two dimensional heat equation

5. Conclusion

On unbounded time intervals the proposed algorithm converges linearly, provided there is a strong enough dissipation mechanism. Thus the new algorithm inherits the classical convergence behavior of the overlapping Schwarz algorithm for elliptic problems, and also the classical convergence behavior of waveform relaxation on long time windows.

On bounded time intervals the convergence is superlinear, which is a new result for domain decomposition algorithms. The waveform relaxation theory predicts superlinear convergence, but the presence of diffusion leads to a faster superlinear convergence rate than the one predicted by waveform relaxation theory [6].

Note that the superlinear convergence rates do not depend on the number of subdomains and thus there is no coarse mesh needed to avoid deterioration of the algorithm [2].

References

1. X. C. Cai, *Additive Schwarz algorithms for parabolic convection-diffusion equations*, Numerische Mathematik **60** (1991), 41–61.
2. M.J. Gander, *Analysis of parallel algorithms for time dependent partial differential equations*, Ph.D. thesis, Stanford University, California, USA, 1997.
3. ———, *A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations*, submitted to Numerical Linear Algebra with Applications (1997).
4. M.J. Gander and A.M. Stuart, *Space-time continuous analysis of waveform relaxation for the heat equation*, to appear in SIAM Journal on Scientific Computing (1997).
5. M.J. Gander and H. Zhao, *Overlapping Schwarz waveform relaxation for parabolic problems in higher dimension*, Proceedings of Algoritmy'97, 1997.
6. E. Giladi and H.B. Keller, *Space-time domain decomposition for parabolic problems*, submitted to SINUM (1997).
7. J. Janssen and S. Vandewalle, *Multigrid waveform relaxation on spatial finite-element meshes: The continuous-time case*, SIAM J. Numer. Anal. **33** (1996), no. 6, 456–474.
8. R. Jeltsch and B. Pohl, *Waveform relaxation with overlapping splittings*, SIAM J. Sci. Comput. **16** (1995), 40–49.
9. Y. Kuznetsov, *Domain decomposition methods for unsteady convection-diffusion problems*, Proceedings of the Ninth International Conference in Computing Methods in Applied Sciences and Engineering, 1990, pp. 211–227.
10. Ch. Lubich and A. Ostermann, *Multi-grid dynamic iteration for parabolic equations*, BIT **27** (1987), 216–234.
11. G.A. Meurant, *A domain decomposition method for parabolic problems*, APPLIED NUMERICAL MATHEMATICS **8** (1991), 427–441.

SCIENTIFIC COMPUTING AND COMPUTATIONAL MATHEMATICS, STANFORD UNIVERSITY, STANFORD, CA 94305

Current address: Centre de Mathématiques Appliquées, École Polytechnique, 91128 Palaiseau, CEDEX, France

E-mail address: mgander@cmapx.polytechnique.fr

Domain Decomposition, Operator Trigonometry, Robin Condition

Karl Gustafson

1. Introduction

The purpose of this paper is to bring to the domain decomposition community certain implications of a new operator trigonometry and of the Robin boundary condition as they pertain to domain decomposition methods and theory. In Section 2 we recall some basic facts and recent results concerning the new operator trigonometry as it applies to iterative methods. This theory reveals that the convergence rates of many important iterative methods are determined by the operator angle $\phi(A)$ of A : the maximum angle through which A may turn a vector. In Section 3 we bring domain decomposition methods into the operator trigonometric framework. In so doing a new three-way relationship between domain decomposition, operator trigonometry, and the recently developed strengthened C.B.S. constants theory, is established. In Section 4 we examine Robin–Robin boundary conditions as they are currently being used in domain decomposition interface conditions. Because the origins of Robin’s boundary condition are so little known, we also take this opportunity to enter into the record here some recently discovered historical facts concerning Robin and the boundary condition now bearing his name.

2. Operator Trigonometry

This author developed an operator trigonometry for use in abstract semigroup operator theory in the period 1966–1970. In 1990 [9] the author found that the Kantorovich error bound for gradient methods was trigonometric: $E_A^{1/2}(x_{k+1}) \leq (\sin A)E_A^{1/2}(x_k)$. Later [14] it was shown that Richardson iteration is trigonometric: the optimal spectral radius is $\rho_{\text{opt}} = \sin A$. Many other iterative methods have now been brought into the general operator trigonometric theory: Preconditioned conjugate gradient methods, generalized minimum residual methods, Chebyshev methods, Jacobi, Gauss–Seidel, SOR, SSOR methods, Uzawa methods and AMLI methods. Also wavelet frames have been brought into the operator trigonometric theory, see [6]. The model (Dirichlet) problem has been worked out in some detail to illustrate the new operator trigonometryc theory, see [5]. There ADI methods are also brought into the operator trigonometric theory. For full information about

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 47B44, 01A65.

this new operator trigonometry, we refer the reader to [9, 11, 10, 12, 14, 13, 18] from which all needed additional details may be obtained.

For our purposes here it is sufficient to recall just a few salient facts. The central notion in the operator trigonometry theory is the angle of an operator, first defined in 1967 through its cosine. Namely, the angle $\phi(A)$ in the operator trigonometry is defined for an arbitrary strongly accretive operator A in a Banach space by

$$(1) \quad \cos A = \inf \frac{\operatorname{Re} \langle Ax, x \rangle}{\|Ax\| \|x\|}, \quad x \in \mathcal{D}(A), \quad Ax \neq 0.$$

For simplicity we may assume in the following that A is a SPD matrix. By an early (1968) min-max theorem the quantity $\sin A = \inf_{\epsilon > 0} \|\epsilon A - I\|$ enjoys the property $\cos^2 A + \sin^2 A = 1$. For A a SPD matrix we know that

$$(2) \quad \cos A = \frac{2\lambda_1^{1/2}\lambda_n^{1/2}}{\lambda_n + \lambda_1}, \quad \sin A = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}.$$

3. Domain Decomposition

Here we will establish the general relationship between domain decomposition methods and the operator trigonometry by direct connection to the treatments of domain decomposition in [19, 3, 23, 21, 2, 20], in that order. Proofs and a more complete treatment will be given elsewhere [4].

We turn first to the treatment of domain decomposition methods in [19, Chapter 11]. Using a similar notation, let $A > 0$, $W_\chi \cong A_\chi > 0$, assume there exist upper and lower bounds C^{upper} and C_{lower} such that the decomposition, $x = \sum_J p_\chi x^\chi$ exists for every $x \in X$ and such that

$$(3) \quad \frac{1}{\Gamma} = C_{\text{lower}} \leq \frac{\sum_J \langle Ap_\chi x^\chi, x^\chi \rangle}{\langle Ax, x \rangle} \equiv \frac{\sum_j \|p_\chi x^\chi\|_A^2}{\|x\|_A^2} \leq C^{\text{upper}} = \frac{1}{\gamma}.$$

Here γ and Γ are the optimal bounds in $\gamma W^{\text{addSI}} \leq A \leq \Gamma W^{\text{addSI}}$ i.e., the condition number $\kappa(W^{-1}A)$ is the ratio Γ/γ . Then it follows for two nonoverlapping domains that the optimal convergence spectral radius is $\rho(M_{\theta_{\text{optimal}}}^{\text{addSI}}) = \|M_{\theta_{\text{optimal}}}^{\text{addSI}}\|_A \leq \frac{\Gamma-\gamma}{\Gamma+\gamma}$. When conjugate gradient is applied to the additive Schwarz domain decomposition algorithm, in the two level case the asymptotic convergence rate improves to $\rho(CGM_{\theta_{\text{optimal}}}^{\text{addSI}}) = \delta/1 + \sqrt{1 - \delta^2}$ where

$$(4) \quad \delta = \sup_{x \in \mathcal{R}(p_1), y \in \mathcal{R}(p_2)} \frac{\langle x, y \rangle_A}{\|x\|_A \|y\|_A}$$

is the C.B.S. constant associated with the two-level decomposition.

THEOREM 1. [4]. *Under the stated conditions, the optimal convergence rate of the additive Schwarz domain decomposition algorithm is trigonometric: $\rho(M_{\theta_{\text{optimal}}}^{\text{addSI}}) = \sin((W^{-1/2}AW^{-1/2})^{1/2})$. In the two level case with the conjugate gradient scheme applied, the optimal asymptotic convergence rate is also trigonometric: $\rho(CGM_{\theta_{\text{optimal}}}^{\text{addSI}}) = \sin((W^{-1/2}AW^{-1/2})^{1/2})$.*

We may obtain an abstract version of Theorem 1 by following the abstract treatment of [3].

THEOREM 2. [4]. *With R and R^* the embedding (restriction) and conjugate (prolongation) operators for $\tilde{V} = V_0 \times V_1 \times \cdots \times V_J$ and B defined by the Fictitious Subspace Lemma as in [3], $\rho_{\text{addSI}}^{\text{optimal}} = \sin(RB^{-1}R^*A)$.*

Next we comment on an important connection between the operator trigonometry and the C.B.S. constants which is inherent in the above results. Turning to [23], for the preconditioned system BA with $BSPD$ and $\rho \equiv \|I - BA\|_A < 1$, one knows that $\kappa(BA) \leq \frac{1+\rho}{1-\rho}$. When optimized, the preconditioned Richardson iteration $u^{k+1} = u^k + \omega B(f - Au^k)$ has error reduction rate $(\kappa(BA) - 1)/(\kappa(BA) + 1)$ per iteration. This means the error reduction rate is exactly $\sin(BA)$, [14, Theorem 5.1] applying here since BA is SPD in the A inner product. From these considerations we may state

THEOREM 3. [4]. *Under the above conditions for the preconditioned system ωBA , the spectral radius $\rho \equiv \|I - \omega BA\|_A$ plays the role of strengthened C.B.S. constant when $\omega = \omega^*$ optimal.*

The principle of Theorem 3 could be applied to the whole abstract theory [21], i.e., to additive multilevel preconditionings $BA = \sum_{i=1}^p T_i$ and multiplicative preconditionings $BA = I - \sum_{i=0}^p (I - T_{p-i})$. The relationships of this principle to the Assumptions 1, 2, 3 of the domain decomposition theory are interesting, inasmuch as the three constants $c_0, \rho(\mathcal{E})$, and ω of those three assumptions are closely related to $\lambda_{\max}(BA)$ and $\lambda_{\min}(BA)$, viz. $c_0^{-2} \leq \lambda_{\min}(BA) \leq \dots \leq \lambda_{\max}(BA) \leq \omega[1 + \rho(\mathcal{E})]$.

Next we turn to operator trigonometry related to the FETI (Finite Element Tearing and Interconnecting) algorithm [2]. See [8] where in an early paper we discussed the potential connections between Kron's tearing theories and those of domain decomposition and FEM and where we utilize graph-theoretic domain decomposition methods to decompose finite element subspaces according to the Weyl–Helmholtz–Hodge parts, which permits the computation of their dimensions. The FETI algorithm is a nonoverlapping domain decomposition with interfacing represented by Lagrange multipliers. To establish a connection of FETI to the operator trigonometry, let us consider the recent [20] analysis of convergence of the FETI method. There it is shown that the condition number of the preconditioned conjugate gradient FETI method is bounded independently of the number of subdomains, and in particular, that

$$(5) \quad \kappa = \frac{\lambda_{\max}(P_V M P_V F)}{\lambda_{\min}(P_V M P_V F)} \leq \frac{c_2 c_4}{c_1 c_3} \leq C \left(1 + \log \frac{H}{h} \right)^\gamma$$

where $P_V F$ is the linear operator of the dual problem of interest and where $P_V M$ is its preconditioner, c_1 and c_2 are lower and upper bounds for F , c_3 and c_4 are lower and upper bounds for M , and where $\gamma = 2$ or 3 depending on assumptions on the FEM triangulation, h being the characteristic element size, H an element-mapping-Jacobian bound.

THEOREM 4. [4] *The operator angle of the FETI scheme is bounded above according to $\sin \phi(P_V M P_V F) \leq (C(1 + \log \frac{H}{h})^\gamma - 1)/2$.*

4. Robin Condition

To reduce overlap while maintaining the benefits of parallelism in Schwarz alternating methods, in [22] a generalized Schwarz splitting including Robin type interface boundary conditions was investigated. Certain choices of the coefficients in the Robin condition were found to lead to enhanced convergence. In the notation of [22] the Robin conditions are written $g_i(u) = \omega_i u + (1 - \omega_i) \frac{\partial u}{\partial n}$, $i = 1, 2$, for two overlapping regions. Later in the discretizations another coefficient α is introduced,

α related to ω by the relation $\omega = (1 - \alpha)/(1 - \alpha + h\alpha)$, h the usual discretization parameter. The case $\omega = 1$ ($\alpha = 0$) corresponds to Dirichlet interface condition, the case $\omega = 0$ ($\alpha = 1$) corresponds to Neumann interface condition. Optimal convergence rates are found both theoretically and for computational examples for α in values near 0.9, i.e., for ω near $1/(1 + 9h)$.

Let us convert these coefficients to the standard Robin notation [7]

$$(6) \quad \frac{\partial u}{\partial n} + \alpha_R u = f$$

where we have used α_R for the Robin coefficient to avoid confusion with the α of [22]. Then in terms of the ω of [22] the Robin condition there becomes $\frac{\partial u}{\partial n} + \omega(1 - \omega)^{-1}u = f$ where the right hand side has absorbed a factor $(1 - \omega)^{-1}$. For the successful $\alpha \approx 0.9$ of [22] the Robin constant $\alpha_R \approx 1/9h$. Since h was relatively small in the simulations of [22], this means that α_R was relatively large and that the Robin condition employed there was “mostly Dirichlet.” This helps intuition, noted in [22] as lacking.

Next let us examine the theoretical analysis of [22]. Roughly, one wants to determine the spectral radius of a block Jacobi matrix $J = M^{-1}N$. The route to do so travels by similarity transformations from J to \tilde{J} to G to G' to $HG'H$ to elements of the last columns of T_1^{-1} and T_2^{-1} to four eigenvalues $\lambda_{1,2,3,4}$ and hence to the spectral radius of J . How have the Robin interface conditions affected the spectral radii of J depicted in the figures of [22]? To obtain some insight let us consider the following simple example. The one dimensional problem $-u'' = f$ is discretized by centered differences over an interval with five grid points x_0, \dots, x_4 , with a Robin boundary condition (6) at x_0 and x_4 . For the three interior unknown values u_1, u_2, u_3 we then arrive at the matrix equation

$$(7) \quad \begin{bmatrix} 2 - \beta & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 - \beta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} ch\beta + f_1 \\ f_2 \\ dh\beta + f_3 \end{bmatrix}$$

where we have absorbed the left Robin boundary condition $-(u_1 - u_0)/h + \alpha_R u_0 = c$ and the right Robin boundary condition $(u_4 - u_3)/h + \alpha_R u_4 = d$, and where β denotes $(1 + \alpha_R h)^{-1}$. Then the matrix A of (7) has eigenvalues $\lambda_1 = 2 - \beta/2 - (\beta^2 + 8)^{1/2}/2$, $\lambda_2 = 2 - \beta$, $\lambda_3 = 2 - \beta/2 + (\beta^2 + 8)^{1/2}/2$. For the successful $\alpha = 0.9$ and assuming $h = 1/45$ (corresponding to [22]) we find $\beta = 0.9$ and hence $\lambda_1 = 0.066$, $\lambda_2 = 1.1$, $\lambda_3 = 3.034$. Using Dirichlet rather than Robin boundary conditions corresponds to $\beta = 0$ and $\lambda_1 = 0.586$, $\lambda_2 = 2$, $\lambda_3 = 3.414$. The condition numbers are $\kappa_{\text{Robin}} \cong 45.97$ and $\kappa_{\text{Dirichlet}} \cong 5.83$. The Robin condition moves the spectrum downward and increases condition number. The worst case approaches Neumann and infinite condition number. One needs to stay mostly Dirichlet: this is the intuition. Also the size of the grid parameter h is critical and determines the effective Robin constant.

Turning next to [1] and the Robin–Robin preconditioner techniques employed there, we wish to make two comments. First, for the advection–diffusion problems $Lu = cu + \vec{a} \cdot \nabla u - \nu \Delta u = f$ being considered in [1], the extracted Robin–Robin interface conditions

$$(8) \quad \left(\nu \frac{\partial}{\partial n_k} - \frac{\vec{a} \cdot \vec{n}_k}{2} \right) v_k = g_k$$

is really just an internal boundary trace of the differential operator and therefore is not independent in any way. A better terminology for (8) might be Peclet–Peclet, corresponding to the well-known Peclet number $P = \frac{aL}{\nu}$ over a fluid length L . Second, when the sign of \vec{a} may change with upwinding or downwinding, the coefficient α_R in (6) may become an eigenvalue in the boundary operator. Solution behavior can then become quite different. Such an internal Steklov–Steklov preconditioner would permit interior “flap” of solutions.

Robin was (Victor) Gustave Robin (1855–1897) who lectured at the Sorbonne at the end of the previous century. Our interest in Robin began twenty years ago when writing the book [7]. The results of the subsequent twenty year search will be published now in [15, 16] to commemorate the 100th anniversary of Robin’s death in 1897. Little is known about Robin personally. However we have uncovered all of his works (they are relatively few). Nowhere have we found him using the Robin boundary condition. Robin wrote a nice thesis in potential theory and also worked in thermodynamics. We have concluded that it is neither inappropriate nor especially appropriate that the third boundary condition now bears his name.

5. Conclusion

In the first domain decomposition conference [17] we presented new applications of domain decomposition to fluid dynamics. Here in the tenth domain decomposition conference we have presented new theory from linear algebra and differential equations applied to domain decomposition.

References

1. Y. Achdou and F. Nataf, *A Robin–Robin preconditioner for an advection–diffusion problem*, C.R. Acad. Sci. Paris (To appear), also the presentation by Y. Achdou at this conference.
2. C. Farhat and F. Roux, *Implicit parallel processing in structural mechanics*, Computational Mechanics Advances (Amsterdam) (J. Oden, ed.), vol. 2, North Holland, 1994, pp. 1–124.
3. M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math. **70** (1995), 163–180.
4. K. Gustafson, *Operator trigonometry of domain decomposition*, To appear.
5. ———, *Operator trigonometry of the model problem*, To appear.
6. ———, *Operator trigonometry of wavelet frames*, To appear.
7. ———, *Partial differential equations and Hilbert space methods*, Wiley, N.Y., 1980, 1987, Dover, NJ, 1997.
8. ———, *Principles of electricity and economics in fluid dynamics*, Num. Meth. Partial Diff. Eqns. **2** (1985), 145–157.
9. ———, *Antieigenvalues in analysis*, Proceedings of the Fourth International Workshop in Analysis and its Applications (Novi Sad, Yugoslavia) (C. Stanojevic and O. Hadzic, eds.), 1991, pp. 57–69.
10. ———, *Antieigenvalues*, Linear Algebra and its Applications **208/209** (1994), 437–454.
11. ———, *Operator trigonometry*, Linear and Multilinear Algebra **37** (1994), 139–159.
12. ———, *Matrix trigonometry*, Linear Algebra and its Applications **217** (1995), 117–140.
13. ———, *Lectures on computational fluid dynamics, mathematical physics, and linear algebra*, Kaigai Publishers, Tokyo, Japan, 1996, World Scientific, Singapore, 1997.
14. ———, *Operator trigonometry of iterative methods*, Num. Lin. Alg. with Appl. **4** (1997), 333–347.
15. K. Gustafson and T. Abe, *The third boundary condition—was it Robin’s?*, The Mathematical Intelligencer **20** (1998), 63–71.
16. ———, *Victor Gustave Robin: 1855–1897*, The Mathematical Intelligencer **20** (1998), To appear.

17. K. Gustafson and R. Leben, *Vortex subdomains*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (R. Glowinski, G. Golub, G. Meurant, and J. Periaux, eds.), SIAM, 1988, pp. 370–380.
18. K. Gustafson and D. Rao, *Numerical range: The field of values of linear operators and matrices*, Springer, New York, 1997.
19. W. Hackbusch, *Iterative solution of large sparse systems of equations*, Springer, Berlin, 1994.
20. J. Mandel and R. Tezaur, *Convergence of a substructuring method with Lagrange multipliers*, Numer. Math. **73** (1996), 473–487.
21. B. Smith, P. Bjorstad, and W. Gropp, *Domain decompositions: Parallel multilevel methods for elliptic partial differential equations*, Oxford Press, Oxford, 1996.
22. W. Tang, *Generalized Schwarz splittings*, SIAM J. Sci Stat. Comput. **13** (1992), 573–595.
23. J. Xu, *Iterative methods by subspace decomposition and subspace correction*, SIAM Review **34** (1992), 581–613.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF COLORADO, BOULDER, CO 80309–0395
E-mail address: gustafs@euclid.colorado.edu

On Orthogonal Polynomial Bases for Triangles and Tetrahedra Invariant under the Symmetric Group

Gary Man-Kwong Hui and Howard Swann

1. Introduction

We present an L_2 -orthonormal polynomial basis for triangles containing 10th degree polynomials in its span. The sixty-six basis functions are defined by using 35 generating functions $\{B_k(x, y)\}$ with the property that $B_k(x, y)$ is orthogonal to $B_k(y, x)$ unless they are equal. For tetrahedra, we describe methods for constructing a an L_2 -orthonormal basis by defining generating functions $B_k(x, y, z)$ such that the action of S_3 on the arguments of B_k can provide as many as six orthogonal basis functions. Thirty-five basis functions generated by 11 B_k have been computed. These bases are particularly useful for approximating the solution of partial differential equations using the Cell Discretization Algorithm (CDA).

The CDA allows a user to partition the domain of a problem into ‘cells’, choose any basis on each cell, and then ‘glue’ the finite dimensional approximations on each cell together across cell interfaces to achieve a form of weak continuity using a method called ‘moment collocation.’ This method allows a user to select a basis tailored to the type of an equation or the geometry of the cells without having to worry about continuity of an approximation.

If polynomial bases are used and we have planar interfaces between cells, we can impose sufficient collocation moments so that our approximations are continuous and we duplicate the $h - p$ finite element method [9]. Error estimates that establish convergence of the method contain two components; the first consists of terms arising from the lack of continuity of an approximation and the second contains terms majorized by the orthogonal complement of the projection of the solution onto the approximation space. However, in all trials of the method using polynomial bases[4, 9, 5, 7, 6, 8], there has been no particular advantage in enforcing continuity of an approximation; continuity does eliminate the first error component, but by doing so a parameter in the second error component grows strongly, thus cancelling any apparent gain by forcing continuity. This is discussed extensively in [9]. Thus we obtain additional degrees of freedom that can, for example, be used to

1991 *Mathematics Subject Classification.* Primary 41A10; Secondary 65N30, 35C10.

Key words and phrases. Orthogonal polynomial basis, tetrahedra, partial differential equations, Galerkin method.

enforce a weak solenoidal condition for approximating the solutions to the Stokes equations [8].

We have implemented the algorithm for general domains in \mathbf{R}^2 partitioned into cells with linear internal interfaces between cells. Affine transformations are used to map any cell into a standard configuration to effect quadrature and if a basis is defined on a cell in standard configuration, we use an affine transformation to provide a basis for the affine image of the cell. For the most part we use cells that are parallelograms or triangles—affine images of a unit square or unit simplex. A ‘good’ basis for a general implementation of the algorithm is a basis that is L_2 orthonormal, particularly for time-dependent problems and the construction of a solenoidal basis [8]. Since affine transformations preserve orthogonality, it suffices to construct orthonormal bases for the standard square or simplex. A global orthonormal basis is then produced by a linear combination of the cell basis functions using coefficients obtained from the QR decomposition of the matrix enforcing the moment collocation constraints [5, 7, 6]. These arguments generalize to \mathbf{R}^3 .

Products of Legendre polynomials provide an L_2 - orthonormal basis for a square. In Section 2, for the unit 2-simplex, with vertices at $(0,0)$, $(1,0)$ and $(0,1)$, we describe the method we have used to construct an orthonormal basis with polynomials of degree 10 or less in its span.

In \mathbf{R}^3 ; products of Legendre Polynomials produce an orthonormal basis for any parallelepiped. In Section 3 we describe the construction of an orthonormal basis for the standard 3-simplex. The methods we use require that we solve a set of four simultaneous quadratic equations in five variables.

The results for tetrahedra were obtained by Hui [2].

2. Construction of a polynomial basis for triangles.

We contrive an L_2 orthonormal basis $B_i(x, y)$ for the 2-simplex that uses a method similar to the Gram-Schmidt process to sequentially introduce sets of monomials $x^j y^k$ into the basis. The first problem is to determine how j and k should be successively chosen to produce our basis sequence.

Consider the Taylor’s expansion of any $u(x, y)$ around (x_o, y_o) :

$$\begin{aligned} u(x, y) = & u(x_o, y_o) + u_x(x - x_o) + u_y(y - y_o) + \\ & (1/2!)[u_{xx}(x - x_o)^2 + 2u_{xy}(x - x_o)(y - y_o) + u_{yy}(y - y_o)^2] \\ & +(1/3!)[u_{xxx}(x - x_o)^3 + 3u_{xxy}(x - x_o)^2(y - y_o) + \\ & \quad 3u_{xyy}(x - x_o)(y - y_o)^2 + u_{yyy}(y - y_o)^3] \\ & +(1/4!)[u_{xxxx}(x - x_o)^4 + 4u_{xxxy}(x - x_o)^3(y - y_o) + 6u_{xxyy}(x - x_o)^2(y - y_o)^2 \\ & \quad + 4u_{xyyy}(x - x_o)(y - y_o)^3 + u_{yyyy}(y - y_o)^4] + \dots \end{aligned}$$

With no other information available about $u(x, y)$, the terms containing the mixed partial derivatives in the expansion with coefficients containing factors 2,3,3,4,6,4, 6,4, appear to be more important than those involving $u_{xx}, u_{yy}, u_{xxx}, u_{yyy}$, and so forth.

Polynomial approximation theory suggests that we introduce monomials into the basis span according to increasing degree and, given any chosen degree, the form of the Taylor’s series suggests that the monomials with equal coefficient factors, which are either a pair $\{x^i y^j, x^j y^i\}$ or of form $x^k y^k$, be added to the basis span in order of *decreasing* coefficient factors. Thus, for example, when generating a

basis that spans polynomials of degree 4, we would first introduce monomial x^2y^2 into the basis set (with Maclaurin series coefficient $u_{xxyy}(0,0)6/4!$), then the pair $\{x^3y, xy^3\}$ (with Maclaurin series coefficients $u_{xxxy}(0,0)4/4!$ and $u_{xyyy}(0,0)4/4!$) and finally the pair $\{x^4, y^4\}$. Our method follows this algorithm.

This gives a justification for increasing the number of basis functions used in the approximation gradually, lessening the need for a new full degree basis at each new approximation.

We call a function $f(x, y)$ **symmetric** if $f(x, y) = f(y, x)$; recalling that the 2-simplex is to be our domain for f , the axis of symmetry is the line $y = x$.

We say function f is **skew** if $f(x, y) = -f(y, x)$. The product of two symmetric functions is symmetric; the product of two skew functions is symmetric, and the product of a symmetric function and a skew function is skew. One easily shows that the integral of a skew function over the standard simplex is zero. We combine our monomials to form expressions that are either *symmetric* or *skew* and have the same span; our sequence of generating functions is given by two sets

$$A \equiv \{1, (x+y), xy, (x^2 + y^2), (x^2y + xy^2), (x^3 + y^3), \dots\} \text{ and}$$

$$B \equiv \{(x-y), (x^2 - y^2), (x^2y - xy^2), (x^3 - y^3), \dots\}.$$

If we integrate by parts over the standard triangle and use a recursive argument, we obtain

$$\int_0^1 \int_0^{1-x} x^p y^q dy dx = [p!q!]/(p+q+2)!.$$

This gives us an exact (rational) value for the $L_2(\text{simplex})$ inner product (denoted $\langle \cdot, \cdot \rangle$) of any monomials.

We use an algorithm equivalent to the Gram-Schmidt process to generate a sequence of symmetric orthogonal polynomials $\{Q_1, Q_2, \dots\}$ from generating set A and a set of skew orthogonal polynomials $\{S_1, S_2, \dots\}$ from B .

Our basis is obtained by combining these two sets using the heuristic suggested by the Maclaurin series. For example, to generate the 7th(and 8th) basis functions, thus introducing x^2y and xy^2 into the basis, we form

$$\alpha \pm \beta \equiv [2 \langle Q_5, Q_5 \rangle]^{-1/2} Q_5 \pm [2 \langle S_3, S_3 \rangle]^{-1/2} S_3.$$

Then symmetric α is orthogonal to skew β and the skew span of B ; skew β is orthogonal to the symmetric span of A ; α and β have norm $1/\sqrt{2}$, so

$$\langle \alpha + \beta, \alpha - \beta \rangle = 1/2 - 1/2 = 0$$

and $\|\alpha + \beta\|^2 = \langle \alpha, \alpha \rangle + \langle \beta, \beta \rangle = 1 = \|\alpha - \beta\|^2$. If $B(x, y) \equiv \alpha + \beta$, then $B(y, x) = \alpha - \beta$.

When generating basis functions with a symmetric lead term, such as 1, xy , x^2y^2 and so forth, where there is no skew partner, we use only the appropriate Q_1, Q_3, Q_7, \dots ; there is no skew β term.

These computations were done with care, for the matrices in the linear systems employed by the Gram-Schmidt process are very ill-conditioned. Our computations were nevertheless exact, for the matrices and vectors are arrays of rational numbers, so that the solution is rational and we have written a program that does Gaussian elimination and back substitution using rational arithmetic, thus keeping control of the instability of the system. A set of 36 polynomials has been computed, producing 66 basis functions, which allow us to generate any polynomial of degree 10 or less.

FORTRAN77 code and the necessary coefficients to generate the full set of basis functions (and their first derivatives) are available from the second author.

The use of this polynomial basis for solving partial differential equations with domains partitioned into triangles requires an efficient method for doing quadrature; points and weights for Gaussian quadrature over triangles, exact for polynomials of degree 20 or less, have been obtained by Dunavant [1]. As in [3], we generate and store an array that contains the information to look up, for example, the computations $\langle \frac{\partial}{\partial x} B_i, \frac{\partial}{\partial y} B_j \rangle$ for use when the partial differential equation has constant coefficients.

3. A symmetric ortho-normal basis for tetrahedra.

The Maclaurin series expansion for $f(x, y, z)$ is

$$\begin{aligned} & f(\mathbf{0}) \\ & + f_x x + f_y y + f_z z \\ & + (1/2)[2(f_{xy}xy + f_{xz}xz + f_{yz}yz) + f_{xx}x^2 + f_{yy}y^2 + f_{zz}z^2] \\ & + (1/6)[6f_{xyz}xyz + 3(f_{xxy}x^2y + \dots + f_{yyx}y^2x + \dots) + f_{xxx}x^3 + \dots] + \\ & (1/24)[12(f_{xxyz}x^2yz + \dots) + 6(f_{xxyy}x^2y^2 + \dots) + 4(f_{xxxy}x^3y + \dots) + \\ & (f_{xxxx}x^4 + \dots)] + \dots \end{aligned}$$

Proceeding naively as before, we assume that, for any particular degree of basis functions, we should initially introduce monomials $x^p y^q z^r$ into the basis that correspond to the larger integer multipliers: 2 then 1; 6,3 then 1; 12,6,4 then 1 and so forth. The monomials that are associated with these multipliers occur in sets of 1 (e.g. $\{xyz\}$), 3 (e.g. $\{xy, xz, zy\}$) or 6 (e.g. $\{x^2y, x^2z, y^2x, y^2z, z^2x, z^2y\}$.) To minimize the number of functions that need to be generated, ideally, our symmetric orthonormal basis would require just one *basis generating* function $B(x, y, z)$ for each of the classes; for the classes with 3 members, $\{B(x, y, z), B(y, z, x), B(z, x, y)\}$ would be an orthonormal set, also orthogonal to the basis functions generated previously; we will call such functions *3-fold basis generating functions*. For the classes with 6 members, the full group S_3 of permutations of $B(x, y, z)$:

$$\{B(x, y, z), B(y, z, x), B(z, x, y), B(y, x, z), B(x, z, y), B(z, y, x)\}$$

would constitute an orthonormal set, orthogonal to the previously generated basis functions; we will call these *6-fold basis generating functions*.

Figure 1 shows a triangular array of the homogeneous monomials of degree 5, with the numbers below each monomial representing the bold-face integer multiplier to be used in the Maclaurin expansion above. For any particular degree, monomials with the same number under them identify those that would be included in the same set as described above. Those with higher numbers would be introduced into the basis first.

Our study takes place in the subspace S of $L_2(3\text{-simplex})$ consisting of polynomials in x, y and z . We denote the inner product $\langle \cdot, \cdot \rangle$.

Each member of the permutation group S_3 induces a linear transformation on S :

If T is the permutation (x, y, z) , it acts on \mathbf{R}^3 as $T \langle x, y, z \rangle = \langle y, z, x \rangle$; $T^2 \langle x, y, z \rangle = T \langle y, z, x \rangle = \langle z, x, y \rangle$; T^3 is the identity. T acts on a polynomial in the following fashion: $T(2x^2yz + 3xz) = T(2x^2y^1z^1 + 3x^1y^0z^1) = 2y^2zx + 3yx$.

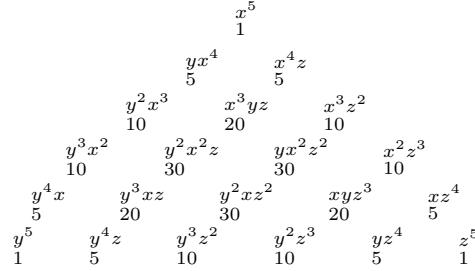


FIGURE 1. Fifth degree monomials.

Transformation $P \equiv P_{xy}$ corresponds to permutation (x, y) . The basic relator is $PT = T^2P$. Since these transformations are to act on any polynomial $q(x, y, z)$, we adopt the convention that, for example, in computing $TP_{xy}q(x, y, z)$, P_{xy} acts first and then T , so the transformation $TP_{xy} = (x, y, z)(x, y) = (x, z)$ and $T^2P_{xy} = (y, z)$. We also use notation S_3 for these transformations.

We let ξ represent a generic member of \mathbf{R}^3 ; given ξ , bold face symbol \mathbf{x}^α represents the monomial $x^i y^j z^k$ associated with any triple of non-negative integers $[\alpha] = [i, j, k]$. For any \mathbf{x}^α , the set of permutations of \mathbf{x}^α is

$$\{\mathbf{x}^\alpha, T\mathbf{x}^\alpha, T^2\mathbf{x}^\alpha, P\mathbf{x}^\alpha, TP\mathbf{x}^\alpha, T^2P\mathbf{x}^\alpha\},$$

where there will be duplicates if the set has only 3 or 1 member. We let T act on ‘powers’ $[\alpha] = [i, j, k]$ by defining $T[\alpha] = T[i, j, k] \equiv [k, i, j]; P[i, j, k] \equiv [j, i, k]$. Then $T\mathbf{x}^\alpha = \mathbf{x}^{T\alpha}, T^2\mathbf{x}^\alpha = \mathbf{x}^{T^2\alpha}$ and so forth.

In figure 1, monomials belonging in the same set (those with the same number below them) correspond to all permutations of such a triple $[i, j, k]$. If $i = j = k$, there is only one monomial; if two of $\{i, j, k\}$ are the same, there are three monomials in the set; if $\{i, j, k\}$ are all different, there are six.

The integral of monomial $x^p y^q z^r$ over the standard 3-simplex can be shown to be $[p!q!r!]/[(p+q+r+3)!]$ using recursive methods similar to those described above [2]. The value of this integral is invariant under the action of S_3 on the monomials. This symmetry means that the integral depends only on set $\{p, q, r\}$. This observation, together with bilinearity of the inner product, can be used to prove the following lemma:

LEMMA 1. *For any polynomials G and $H \in S$,*

1. *for any $R \in S_3$, $\langle RG, RH \rangle = \langle G, H \rangle$;*
2. *for any $R \in S_3$, $\langle RG, H \rangle = \langle G, R^{-1}H \rangle$;*
3. $\langle G, TH \rangle = \langle T^2G, H \rangle = \langle TG, T^2H \rangle$.

Results 2 and 3 follow readily from 1; 2 shows that the members of S_3 act as unitary operators on S .

Our first basis member is the normalized constant function $B_1 \equiv \sqrt{6}$. When only one basis function is produced, we call these *one-fold generators*. The images under S_3 of the next three basis-generating functions are to contain $\{x, y, z\}$, then $\{xy, yz, xz\}$ and finally $\{x^2, y^2, z^2\}$ in their span.

We give some necessary conditions for recursively defining basis-generating functions B_{r+1} that produce three basis members under the action of T , as is

the case here. Assume appropriate functions $B_k(\xi)$ have already been constructed, $k = 1, \dots, r$.

LEMMA 2. Suppose $[\alpha] = [i_1, i_2, i_3]$ has exactly two of $\{i_1, i_2, i_3\}$ equal. The next function G is expressed as

$$(1) \quad G(\xi) = H(\mathbf{x}^\alpha) + \sum_{k=1}^r \sum_{i=0}^{n(k)} \sum_{j=0}^{m(k)} a_{k,i,j} T^i P^j B_k(\xi)$$

where $n(k) \leq 2$; $m(k) \leq 1$. When $B_k(\xi)$ is a 3-fold basis generating function, $n(k) = 2$ and $m(k) = 0$. Function $H(\mathbf{x}^\alpha) \equiv b_0 \mathbf{x}^\alpha + b_1 T \mathbf{x}^\alpha + b_2 T^2 \mathbf{x}^\alpha$. Suppose

- (I) G, TG , and T^2G are orthogonal to the previous basis functions;
- (II) G, TG and T^2G are pairwise orthogonal and
- (III) set $\{PG, PTG, PT^2G\} = \{G, TG, T^2G\}$.

Then, without loss of generality, the following assumptions can be made about $G, [\alpha], \{b_i\}$ and $\{a_{k,i,j}\}$:

- (a) $PG = G; PH = H$.
- (b) $[\alpha] = [i_1, i_1, i_3]$; the first two powers are equal and $b_1 = b_2$.
- (c) (I) holds if and only if $a_{k,i,j} = -\langle H, T^i P^j B_k \rangle$. Thus the $a_{k,i,j}$ are linear combinations of b_0 and b_1 .
- (d) In view of (a), arguing recursively, without loss of generality, we can assume that all three-fold basis generators B_k satisfy $PB_k = B_k$. Then

$$\text{if } n(k) = 2 \text{ and } m(k) = 0, a_{k,1,0} = a_{k,2,0}.$$

$$\text{If } n(k) = 2 \text{ and } m(k) = 1, a_{k,0,1} = a_{k,0,0}; a_{k,1,1} = a_{k,2,0} \text{ and}$$

$$a_{k,2,1} = a_{k,1,0}.$$

- (e) If the substitutions in (c) and (d) are made, (I), (II) and (III) hold if and only if $\langle G, TH \rangle = 0$.

PROOF. (a) From (III) it follows that exactly one of $\{G, TG, T^2G\}$ must be fixed under P . For example, suppose $PG = T^2G$. Then TG is fixed under P , for $PTG = T^2PG = T^2T^2G = TG$. Now

$$\begin{aligned} TG(\xi) &= TH(\mathbf{x}^\alpha) + \sum_{k=1}^r \sum_{i=0}^{n(k)} \sum_{j=0}^{m(k)} a_{k,i,j} T^{i+1} P^j B_k(\xi) \\ &= H(\mathbf{x}^{T\alpha}) + \sum_{k=1}^r \sum_{i=0}^{n(k)} \sum_{j=0}^{m(k)} a_{k,i,j} T^{i+1} P^j B_k(\xi). \end{aligned}$$

By re-labelling the $a_{k,i,j}$ and defining $[\beta] \equiv T[\alpha]$, this has the same form as (1); call it \tilde{G} . We are assuming that $PTG = TG$; thus $PG = \tilde{G}$. Then $PH(\mathbf{x}^\beta) = H(\mathbf{x}^\beta)$ follows immediately. Redefine \tilde{G} to be G .

- (b) Expanding $H(\mathbf{x}^\beta) = PH(\mathbf{x}^\beta)$ we get $H(\mathbf{x}^\beta) = b_0 \mathbf{x}^\beta + b_1 T \mathbf{x}^\beta + b_2 T^2 \mathbf{x}^\beta$
 $= b_0 \mathbf{x}^\beta + b_1 \mathbf{x}^{T\beta} + b_2 \mathbf{x}^{TT\beta} = PH(\mathbf{x}^\beta) \equiv b_0 P \mathbf{x}^\beta + b_1 PT \mathbf{x}^\beta + b_2 PT^2 \mathbf{x}^\beta$
 $= b_0 \mathbf{x}^{P\beta} + b_1 \mathbf{x}^{PT\beta} + b_2 \mathbf{x}^{PTT\beta}.$

Recalling that $[\beta] = [i_1, i_2, i_3]$ has exactly two of i_1, i_2, i_3 equal, one of $[\beta], T[\beta]$ and $T^2[\beta]$ has these two equal integers in the first two positions and hence this triple is invariant under P . For example, suppose that $PT[\beta] =$

$T[\beta]$. Then $PT^2[\beta] = PTPT[\beta] = [\beta]$ and $P[\beta] = T^2[\beta]$; the assumption that $PH = H$ then requires that $b_0 = b_2$. If we let $[\gamma] \equiv T[\beta]$ and express $H(\mathbf{x}^\gamma)$ as $b_1\mathbf{x}^{T\beta} + b_2T\mathbf{x}^{T\beta} + b_0T^2\mathbf{x}^{T\beta} = b_1\mathbf{x}^\gamma + b_2T\mathbf{x}^\gamma + b_0T^2\mathbf{x}^\gamma$ we get the correct representation by relabelling the b_i 's.

- (c) This follows if we take the inner product of $T^i P^j B_k$ with (1).
- (d) When $m(k) = 0$, the assumption that $PB_k = B_k$ and $PG = G$ readily give the first result. When $m(k) = 1$, we have, for example, $-a_{k,1,1} = \langle H, TPB_k \rangle = \langle T^2H, PB_k \rangle = \langle PT^2H, B_k \rangle = \langle TPH, B_k \rangle = \langle TH, B_k \rangle = \langle H, T^2B_k \rangle = -a_{k,2,0}$.
- (e) Since $\langle G, TG \rangle = \langle TG, T^2G \rangle = \langle T^2G, G \rangle$, pairwise orthogonality follows if we can establish that just one of these is zero. If the substitutions in (c) are made, G will be orthogonal to all $T^i P^j B_k$ for any choice of H , hence orthogonal to the sums in the representation (1) for G . Thus $\langle G, TG \rangle = \langle G, TH \rangle$. The representations in (d) give us (III).

□

This lemma shows that all we need to do to establish the existence of a suitable G is to find some $H(\mathbf{x}^\alpha)$ of form $b_0\mathbf{x}^\alpha + b_1(T\mathbf{x}^\alpha + T^2\mathbf{x}^\alpha)$ so that, when the substitutions in (c) are made, which are linear in $\{b_0, b_1\}$, the expression $\langle G, TH \rangle = 0$ has a real solution. This is a quadratic equation in $\{b_0, b_1\}$. If we first seek only this orthogonality, there really is only one degree of freedom here; we can set b_0 or $b_1 = 1$ so that the requirement that $\langle G, TH \rangle = 0$ yields a quadratic equation in one variable. Any real root gives a suitable G with the orthogonality properties; it's final definition is found by normalizing so that $\langle G, G \rangle = 1$.

The first four basis generators we have computed are

$$\begin{aligned} B_1 &= \sqrt{6}; \\ B_2 &= \sqrt{30}(2(x+y)-1); \\ B_3 &= \sqrt{7/6}(78xy+6z(x+y)-2z-14(x+y)+3); \\ B_4 &= \sqrt{182+56\sqrt{10}}\left((6\sqrt{10}-20)z^2+\sqrt{10}(x^2+y^2)+(2\sqrt{10}-1)xy+(6\sqrt{10}-17)z(x+y)\right. \\ &\quad \left.+(3-2\sqrt{10})(x+y)+(19-6\sqrt{10})z+(\sqrt{10}-5/2)\right). \end{aligned}$$

One-fold basis generators, like the one with lead term xyz , are easily computed. These are to be invariant under S_3 ; for any k , all $a_{k,i,j}$ will be equal. For example, $B_5 = \sqrt{2}(504xyz - 63(xy+yz+xz) + 9(x+y+z) - 3/2)$.

B_6 is the first 6-fold generating function. The lead term is a linear sum of $\{yz^2, zx^2, xy^2, xz^2, yx^2, zy^2\}$. It will have representation

$$G(\xi) = H(\mathbf{x}^\alpha) + \sum_{k=1}^r \sum_{i=0}^{n(k)} \sum_{j=0}^{m(k)} a_{k,i,j} T^i P^j B_k(\xi)$$

as before, except this time all three integers in $[\alpha]$ are different; $[\alpha] = [0, 1, 2]$ in this case, and

$$H(\mathbf{x}^\alpha) = b_0\mathbf{x}^\alpha + b_1T\mathbf{x}^\alpha + b_2T^2\mathbf{x}^\alpha + b_3P\mathbf{x}^\alpha + b_4TP\mathbf{x}^\alpha + b_5T^2P\mathbf{x}^\alpha.$$

We wish to find values for $a_{k,i,j}$ and b_p such that

$$Q \equiv \{G, TG, T^2G, PG, TPG, T^2PG\}$$

is a set of pairwise orthogonal functions, orthogonal to the previous basis functions.

First note that for the functions in Q to be orthogonal to basis functions $T^i P^j B_k(\xi)$ it suffices to show that they are orthogonal to B_k , since Q is to be invariant under S_3 and the adjoints of operators in S_3 are in S_3 .

Next, since the previous basis functions are assumed to be orthonormal, for any B_k , we can make the following reductions with the help of lemma 2.1. The orthogonality requirements are

$$\begin{aligned} 0 &= \langle G, B_k \rangle = \langle H, B_k \rangle + a_{k,0,0} \\ 0 &= \langle TG, B_k \rangle = \langle G, T^2 B_k \rangle = \langle H, T^2 B_k \rangle + a_{k,2,0} \\ 0 &= \langle T^2 G, B_k \rangle = \langle G, TB_k \rangle = \langle H, TB_k \rangle + a_{k,1,0} \\ 0 &= \langle PG, B_k \rangle = \langle G, PB_k \rangle = \langle H, PB_k \rangle + a_{k,0,1} \\ 0 &= \langle TPG, B_k \rangle = \langle G, PT^2 B_k \rangle = \langle H, TPB_k \rangle + a_{k,1,1} \\ 0 &= \langle T^2 PG, B_k \rangle = \langle G, PTB_k \rangle = \langle H, T^2 PB_k \rangle + a_{k,2,1}. \end{aligned}$$

For three-fold generators, where $PB_k = B_k$, the last three requirements are omitted; the associated $a_{k,i,1}$ are zero. In this way we express the $a_{k,i,j}$ as linear combinations of the $\{b_i\}$.

Finally, we need $\{b_i\}$ so that Q is a pairwise orthogonal set. Again, using adjoints, the fifteen requirements reduce to the following four.

$$\begin{aligned} 0 &= \langle G, TG \rangle = \langle G, T^2 G \rangle = \langle TG, T^2 G \rangle = \langle PG, TPG \rangle = \langle PG, T^2 PG \rangle \\ &= \langle TPG, T^2 PG \rangle \text{ (Type 1)} \\ 0 &= \langle G, PG \rangle = \langle TG, TPG \rangle = \langle T^2 G, T^2 PG \rangle \text{ (Type 2)} \\ 0 &= \langle G, TPG \rangle = \langle TG, T^2 PG \rangle = \langle T^2 G, PG \rangle \text{ (Type 3)} \\ 0 &= \langle G, T^2 PG \rangle = \langle TG, PG \rangle = \langle T^2 G, TPG \rangle \text{ (Type 4)}. \end{aligned}$$

If the substitutions for the $a_{k,i,j}$ are made, the members of Q are orthogonal to the previous basis functions, and the four equations above give us four simultaneous quadratic equations in the variables

$$\{b_0, b_1, b_2, b_3, b_4, b_5\}.$$

For example, for B_6 , where all $a_{k,i,1} = 0$ with $k < 6$ and we let $a_{k,i}$ denote $a_{k,i,0}$, the four types above are equivalent to the following, where when $k = 1$ or 5, there is only a single term in the sum.

Type 1.

$$0 = \langle G, TG \rangle = \langle G, TH \rangle = \langle H, TH \rangle - \sum_{k=1}^5 (a_{k,0} a_{k,1} + a_{k,0} a_{k,2} + a_{k,1} a_{k,2})$$

Type 2.

$$0 = \langle G, PG \rangle = \langle G, PH \rangle = \langle H, PH \rangle - \sum_{k=1}^5 (a_{k,0}^2 + 2a_{k,1} a_{k,2})$$

Type 3.

$$0 = \langle G, TPG \rangle = \langle G, TPH \rangle = \langle H, TPH \rangle - \sum_{k=1}^5 (a_{k,2}^2 + 2a_{k,0} a_{k,1})$$

Type 4.

$$0 = \langle G, T^2 PG \rangle = \langle G, T^2 PH \rangle = \langle H, T^2 PH \rangle - \sum_{k=1}^5 (a_{k,1}^2 + 2a_{k,0} a_{k,2}).$$

The normalization requirement is $1 = \langle G, G \rangle = \langle G, H \rangle$. $\langle G, G \rangle$ is a non-negative homogeneous quadratic form; thus $\langle G, G \rangle = 1$ places us on the (compact) 5-dimensional surface of an ellipsoid in \mathbf{R}^6 . We initially confine our attention to fulfilling the orthogonality requirements, so, for example, we can let $b_0 = 1$; we must then find simultaneous roots for 4 quadratic forms in five variables. We use a variant of Newton's method that has proved to be quite effective in obtaining roots rapidly [2].

A number of questions remain.

1. We have computed 11 basis-generating functions so far, which produce the 35 basis functions necessary to have polynomials of the fourth degree or less in their span. More are needed for practical use of this basis. Is there some way of proving that there always exists a solution to the simultaneous quadratics?
2. Assuming that (as is the case in our experiments) there is a one-parameter family of solutions, what criteria should we use for selecting any particular one? We sought solutions \mathbf{b} such that each b_i was about the same magnitude, but with many changes of sign. For example, should we rather choose some solution \mathbf{b} such that just one b_i has a large magnitude?
3. The coefficients become quite large; for example, in B_{11} , with 24 distinct coefficients, the smallest is about 31, the largest about 4890, with 15 greater than 1000. We used double precision Gaussian Quadrature to evaluate all the inner products in the solution algorithm and terminated the algorithm when \mathbf{b} was found so that, for each i , $|f_i(\mathbf{b})| < 10^{-17}$, but tests of orthogonality of the normalized basis functions were beginning to have significant errors, as appears to be the case with such generalizations of the Gram-Schmidt process. The inner products of the monomials are rational; is there a way to exploit this as was done with the basis functions for triangles?

References

1. D.A. Dunavant, *High degree efficient symmetrical Gaussian quadrature rules for the triangle*, Int. J. Num. Meth. in. Engr. **21** (1985), no. 6, 1129–1148.
2. G. M-K. Hui, *On an orthonormal basis for tetrahedra invariant under S_3* , Master's thesis, San Jose State University, 1995.
3. I. N. Katz, A. G. Peano, and M. P. Rossow, *Nodal variables for complete conforming finite elements of arbitrary polynomial order*, Comput. Math. Appl. **4** (1978), no. 2, 85–112.
4. H. Swann, *On the use of Lagrange multipliers in domain decomposition for solving elliptic problems*, Math. Comp. **60** (1993), no. 201, 49–78.
5. ———, *Error estimates using the cell discretization method for some parabolic problems*, Journal of Computational and Applied Mathematics **66** (1996), 497–514.
6. ———, *Error estimates using the cell discretization algorithm for steady-state convection-diffusion equations*, Journal of Computational and Applied Mathematics **82** (1997), 389–405.
7. ———, *Error estimates using the cell discretization method for second-order hyperbolic equations*, Numerical Methods for Partial Differential Equations **13** (1997), 531–548.
8. ———, *On approximating the solution of the stationary Stokes equations using the cell discretization algorithm*, submitted to Numerical Methods for Partial Differential Equations (1998).
9. H. Swann, M. Cayco, and L. Foster, *On the convergence rate of the cell discretization algorithm for solving elliptic problems*, Math. Comp. **64** (1995), 1397–1419.

SAN JOSÉ STATE UNIVERSITY, SAN JOSÉ, CA 95192-0103

E-mail address: swann@mathcs.sjsu.edu

On Schwarz Alternating Methods for Nonlinear Elliptic Problems

Shiu Hong Lui

1. Introduction

The Schwarz Alternating Method is a method devised by H. A. Schwarz more than one hundred years ago to solve linear boundary value problems. It has garnered interest recently because of its potential as a very efficient algorithm for parallel computers. See the fundamental work of Lions in [7] and [8]. The literature on this method for the boundary value problem is huge, see the recent reviews of Chan and Mathew [5] and Le Tallec [14], and the book of Smith, Bjorstad and Gropp [11]. The literature for nonlinear problems is rather sparse. Besides Lions' works, see also Cai and Dryja [3], Tai [12], Xu [15], Dryja and Hackbusch [6], Cai, Keyes and Venkatakrishnan [4], Tai and Espedal [13], and references therein. Other papers can be found in the proceedings of the annual domain decomposition conferences. In this paper, we prove the convergence of the Schwarz sequence for some 2nd-order nonlinear elliptic partial differential equations. We do not attempt to define the largest possible class of problems or give the weakest condition under which the Schwarz Alternating Method converges. The main aim is rather to illustrate that this remarkable method works for a very wide variety of nonlinear elliptic PDEs.

Let Ω be a bounded domain in \mathbf{R}^N with a smooth boundary. Suppose $\Omega = \Omega_1 \cup \Omega_2$, where the subdomains Ω_i have smooth boundaries and are overlapping. We assume the nontrivial case where both subdomains are proper subsets of Ω . Let (u, v) denote the usual $L^2(\Omega)$ inner product and $\|u\|^2 = (u, u)$. Denote the energy inner product in the Sobolev space $H_0^1(\Omega)$ by $[u, v] = \int_{\Omega} \nabla u \cdot \nabla v$ and let $\|u\|_1 = [u, u]^{1/2}$. Denote the norm on $H^{-1}(\Omega)$ by $\|\cdot\|_{-1}$ with

$$\|u\|_{-1} = \sup_{\|v\|_1=1} |[u, v]|.$$

Let Δ_i be the Laplacian operator considered as an operator from $H_0^1(\Omega_i)$ onto $H^{-1}(\Omega_i)$, $i = 1, 2$. The smallest eigenvalue of $-\Delta$ on Ω is denoted by λ_1 while the smallest eigenvalue of $-\Delta_i$ is denoted by $\lambda_1(\Omega_i)$, $i = 1, 2$. The collection of eigenvalues on Ω is denoted by $\{\lambda_j\}_{j=1}^{\infty}$. For notational convenience, we define $\lambda_0 = -\infty$. We take overlapping to mean that $H_0^1(\Omega) = H_0^1(\Omega_1) + H_0^1(\Omega_2)$. In

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65J15.
This work was in part supported by a grant from RGC CERG HKUST726/96E.

this paper, a function in $H_0^1(\Omega_i)$ is considered as a function defined on the whole domain by extension by zero. Let P_i denote the orthogonal (with respect to the energy inner product) projection onto $H_0^1(\Omega_i)$, $i = 1, 2$. It is well known that

$$d \equiv \max(\|(I - P_2)(I - P_1)\|_1, \|(I - P_1)(I - P_2)\|_1) < 1.$$

See Lions [7] and Bramble et. al. [2]. Throughout this paper, C will be used to denote a (not necessarily the same) positive constant.

We shall consider two classes of Schwarz methods. The first class, nonlinear Schwarz method, denotes a method where a sequence of nonlinear problems is solved one subdomain after another one. The second class, linear Schwarz method, is devoted to the method where a linear problem is solved in each subdomain.

The first Schwarz method for nonlinear problems is due to Lions [7]. He considers a functional $I \in C^1(H_0^1(\Omega), \mathbf{R})$ which is coercive, weakly lower semicontinuous, uniformly convex and bounded below. By making a correction alternately in each subdomain which minimizes the functional, he shows that the sequence converges to the unique minimizer of the functional.

2. Nonlinear Schwarz Method

In this section, we use the Schwarz method in conjunction with the methods of Banach and Schauder fixed points and of Global Inversion. The first result is an adaptation of the variational approach of Lions [7] for linear problems to nonlinear problems. We assume the nonlinearity satisfies a certain Lipschitz condition with a sufficiently small Lipschitz constant so that the method of proof for the linear problem still applies. See Lui [9] for a proof.

THEOREM 1. *Consider the equation*

$$(1) \quad -\Delta u = f(x, u, \nabla u) + g \text{ on } \Omega$$

with homogeneous Dirichlet boundary conditions. Assume for every $u, v \in H_0^1(\Omega)$,

$$\|f(x, u, \nabla u) - f(x, v, \nabla v)\| \leq c\sqrt{\lambda_1}\|u - v\|_1,$$

where c is a constant such that $c < 1$ and

$$(2) \quad d < \sqrt{1 - c^2} - c.$$

Assume $g \in L^2(\Omega)$. For $n = 0, 1, 2, \dots$ and some $u^{(0)} \in H_0^1(\Omega)$, define the Schwarz sequence as:

$$-\Delta u^{(n+\frac{1}{2})} = f(x, u^{(n+\frac{1}{2})}, \nabla u^{(n+\frac{1}{2})}) + g \text{ on } \Omega_1, \quad u^{(n+\frac{1}{2})} = u^{(n)} \text{ on } \partial\Omega_1,$$

$$-\Delta u^{(n+1)} = f(x, u^{(n+1)}, \nabla u^{(n+1)}) + g \text{ on } \Omega_2, \quad u^{(n+1)} = u^{(n+\frac{1}{2})} \text{ on } \partial\Omega_2.$$

Then, the Schwarz sequence converges geometrically to the solution of (1) in the energy norm. Here, $u^{(n+\frac{1}{2})}$ is considered as a function in $H_0^1(\Omega)$ by defining it to be $u^{(n)}$ on $\Omega \setminus \Omega_1$ and $u^{(n+1)}$ is defined as $u^{(n+\frac{1}{2})}$ on $\Omega \setminus \Omega_2$.

It is an open problem to determine whether the Schwarz sequence converges geometrically with just the condition $c < 1$.

Next, we give a similar result for an equation whose solution is shown to exist by the Schauder/Schaeffer fixed point theorem. See Nirenberg [10] for instance.

THEOREM 2. Consider the equation

$$(3) \quad -\Delta u = f(x, u, \nabla u) + g \text{ on } \Omega$$

with homogeneous Dirichlet boundary conditions. Assume that $f \in C^1(\overline{\Omega} \times \mathbf{R} \times \mathbf{R}^N)$ and for every $x \in \Omega$, $a \in \mathbf{R}$, $\xi \in \mathbf{R}^N$, $\partial f(x, a, \xi)/\partial u \leq 0$ and $|f(x, a, \xi)| \leq C(1+|\xi|^\gamma)$, where C, γ are positive constants with $\gamma < 1$. Assume $g \in H^1(\Omega)$, $u^{(0)} \in H_0^1(\Omega)$ and sufficiently smooth ($g \in H^{[N/2]+1}(\Omega)$ and $u^{(0)} \in H^{[N/2]+3}(\Omega)$). For $n = 0, 1, 2, \dots$, define the Schwarz sequence as:

$$-\Delta u^{(n+\frac{1}{2})} = f(x, u^{(n+\frac{1}{2})}, \nabla u^{(n+\frac{1}{2})}) + g \text{ on } \Omega_1, \quad u^{(n+\frac{1}{2})} = u^{(n)} \text{ on } \partial\Omega_1,$$

$$-\Delta u^{(n+1)} = f(x, u^{(n+1)}, \nabla u^{(n+1)}) + g \text{ on } \Omega_2, \quad u^{(n+1)} = u^{(n+\frac{1}{2})} \text{ on } \partial\Omega_2.$$

Then, the Schwarz sequence converges geometrically to the solution of (3) in the L^∞ norm.

The proof can be found in Lui [9]. It can be divided into four steps. The first step is to show that the Schwarz sequence is well defined. Next, we show that the sequence is bounded in the H^1 norm so that there exists a weak limit. Then, we show that the sequence actually converges strongly (in H^1) to this limit. Finally, we use the maximum principle to show that this limit is in fact the unique solution to the differential equation. Note that geometric convergence results from the strong maximum principle which is used to show that the ratio of successive errors in the L^∞ norm is bounded by some constant less than one.

It is natural to inquire whether the rather strong condition on the nonlinearity, $\partial f/\partial u \leq 0$, is really necessary. We believe that any restriction on f leading to a unique solution would also do. However, without any conditions on f , the quasilinear equation may have multiple solutions and some numerical evidence suggests that the Schwarz sequence does not converge. We tried several examples for which there are at least two distinct solutions. We monitor $\|u^{(n+\frac{1}{2})} - u^{(n)}\|$ in $\Omega_1 \cap \Omega_2$ and find that it oscillates.

Next, we show that the Schwarz method can be applied to a certain class of semilinear elliptic problem whose solution can be shown to be unique using the Global Inversion Theorem. See Ambrosetti and Prodi [1].

THEOREM 3. Consider the semilinear elliptic equation

$$(4) \quad -\Delta u = \lambda u + f(x, u) + g \text{ on } \Omega$$

with homogeneous Dirichlet boundary conditions. Here $\lambda \in \mathbf{R}$ is given with $\lambda \neq \lambda_j$ for all j and such that there exist positive integers j, l so that for every $t \in \mathbf{R}$,

$$\lambda_{j-1}(\Omega_1) < \lambda + f_u(x, t) \leq \lambda < \lambda_j(\Omega_1), \quad \forall x \in \Omega_1,$$

and

$$\lambda_{l-1}(\Omega_2) < \lambda + f_u(x, t) \leq \lambda < \lambda_l(\Omega_2), \quad \forall x \in \Omega_2.$$

Assume $f \in C^1(\overline{\Omega}, \mathbf{R})$ and satisfies the conditions

$$(5) \quad \frac{\|f(x, v_n)\|_{-1}}{\|v_n\|_1} \rightarrow 0 \quad \text{whenever } \|v_n\|_1 \rightarrow \infty$$

and

$$\lambda_{k-1} < \lambda + f_u(x, t) < \lambda_k$$

for every $x \in \Omega$ and $t \in \mathbf{R}$ and for some $k \in \mathbf{N}$. The function g is assumed to be in $H^{-1}(\Omega)$. For $n = 0, 1, 2, \dots$ and any $u^{(0)} \in H_0^1(\Omega)$, define the Schwarz sequence as:

$$-\Delta u^{(n+\frac{1}{2})} = \lambda u^{(n+\frac{1}{2})} + f(x, u^{(n+\frac{1}{2})}) + g \text{ on } \Omega_1, \quad u^{(n+\frac{1}{2})} = u^{(n)} \text{ on } \partial\Omega_1,$$

$$-\Delta u^{(n+1)} = \lambda u^{(n+1)} + f(x, u^{(n+1)}) + g \text{ on } \Omega_2, \quad u^{(n+1)} = u^{(n+\frac{1}{2})} \text{ on } \partial\Omega_2.$$

Then the Schwarz sequence converges geometrically to the unique solution of the semilinear elliptic equation (4) in the L^∞ norm.

We note that (5) is satisfied when, for instance, f is a bounded function.

For the above semilinear equation, we made use of the property $f_u \leq 0$ in the final step of the proof to show that the limit of the Schwarz sequence is the unique solution to the original problem. It is unknown whether this assumption is really necessary.

Next we consider the resonance problem for the above semilinear equation.

THEOREM 4. Consider the semilinear equation

$$(6) \quad -\Delta u = \lambda_1 u + f(x, u) + g \text{ on } \Omega$$

with homogeneous Dirichlet boundary conditions. Here $f \in C^1(\overline{\Omega}, \mathbf{R})$ and satisfies the following conditions:

1. $\exists M$ such that $|f(x, s)| \leq M$, $\forall x \in \Omega$, $s \in \mathbf{R}$.
2. $\lim_{s \rightarrow \pm\infty} f(x, s) = f_\pm$, $\forall x \in \Omega$.
3. $f_- \cdot \int_\Omega \phi_1 < -\int_\Omega g\phi_1 < f_+ \cdot \int_\Omega \phi_1$, where ϕ_1 is the positive eigenfunction of $-\Delta$ corresponding to the principal eigenvalue λ_1 .
4. $f_u(x, s) \leq 0$, $\forall x \in \Omega$, $s \in \mathbf{R}$.

The function g is assumed to be in $H^{-1}(\Omega)$. For $n = 0, 1, 2, \dots$ and any $u^{(0)} \in H_0^1(\Omega)$, define the Schwarz sequence as:

$$-\Delta u^{(n+\frac{1}{2})} = \lambda_1 u^{(n+\frac{1}{2})} + f(x, u^{(n+\frac{1}{2})}) + g \text{ on } \Omega_1, \quad u^{(n+\frac{1}{2})} = u^{(n)} \text{ on } \partial\Omega_1,$$

$$-\Delta u^{(n+1)} = \lambda_1 u^{(n+1)} + f(x, u^{(n+1)}) + g \text{ on } \Omega_2, \quad u^{(n+1)} = u^{(n+\frac{1}{2})} \text{ on } \partial\Omega_2.$$

Then the Schwarz sequence converges geometrically to the unique solution of the semilinear elliptic equation (6) in the L^∞ norm.

3. Linear Schwarz Method

In the last section, each subdomain problem is still a nonlinear problem. We now consider iterations where linear problems are solved in each subdomain. This is of great importance because in practice, we always like to avoid solving nonlinear problems. One way is in the framework of Newton's method. Write a model semilinear problem as $G(u) = u - \Delta^{-1}f(x, u)$ for $u \in H_0^1(\Omega)$. Suppose it has a solution u and suppose that $\|\Delta^{-1}f_u(x, u)\| < 1$, then for initial guess $u^{(0)}$ sufficiently close to u , the Newton iterates $u^{(n)}$ defined by

$$(7) \quad u^{(n+1)} = u^{(n)} - G_u(u^{(n)})^{-1}G(u^{(n)})$$

converge to u . Note that the assumption means that $G_u = I - \Delta^{-1}f_u$ has a bounded inverse in a neighborhood of u . Now each linear problem (7) can be solved using the classical Schwarz Alternating Method. We take a different approach.

THEOREM 5. Consider the equation

$$(8) \quad -\Delta u = f(x, u, \nabla u) + g \text{ on } \Omega$$

with homogeneous Dirichlet boundary conditions. Assume for every $u, v \in H_0^1(\Omega)$,

$$\|f(x, u, \nabla u) - f(x, v, \nabla v)\| \leq c\sqrt{\lambda_1}\|u - v\|_1,$$

where c is a constant such that $c < 1$ and

$$d < \sqrt{1 - c^2} - c.$$

Assume $g \in L^2(\Omega)$. For $n = 0, 1, 2, \dots$ and any $u^{(0)} \in H_0^1(\Omega)$, define the Schwarz sequence by,

$$-\Delta u^{(n+\frac{1}{2})} = f(x, u^{(n)}, \nabla u^{(n)}) + g \text{ on } \Omega_1, \quad u^{(n+\frac{1}{2})} = u^{(n)} \text{ on } \partial\Omega_1,$$

$$-\Delta u^{(n+1)} = f(x, u^{(n+\frac{1}{2})}, \nabla u^{(n+\frac{1}{2})}) + g \text{ on } \Omega_2, \quad u^{(n+1)} = u^{(n+\frac{1}{2})} \text{ on } \partial\Omega_2.$$

Then, the Schwarz sequence converges to the solution of (8) in the energy norm.

Note that each subdomain problem is a linear one.

4. Work in Progress and Conclusion

We now report some recent progress on Schwarz Alternating Methods for the two-dimensional, steady, incompressible, viscous Navier Stokes equations. In the stream function formulation, these equations reduce to the 4th-order nonlinear elliptic PDE

$$\Delta^2 \psi = RK(\Delta\psi, \psi) + f,$$

where ψ is the stream function, R is the Reynolds number, K is a skew-symmetric bilinear form defined by $K(u, v) = v_y u_x - v_x u_y$, and f is a forcing term. We have constructed three different Schwarz sequences, nonlinear, linear and parallel sequences and have been able to show global convergence of the sequences in the H^2 norm to the true solution provided the Reynolds number is sufficiently small. Here, nonlinear and linear sequences refer to whether nonlinear or linear problems are solved in each subdomain, and parallel sequence refers to the independence of the problems in each subdomain. We give some further details for the nonlinear Schwarz sequence below.

In the general case, the boundary conditions are inhomogeneous. We make a simple change of variable so that the boundary conditions become homogeneous. The problem now becomes $\Delta^2 \phi = RG(\phi)$ where $\phi \in H_0^2(\Omega)$ and G is an appropriate nonlinear term. Let $\phi^{(0)} \in H_0^2(\Omega)$. For $n = 0, 1, 2, \dots$, define the nonlinear Schwarz sequence as

$$\begin{aligned} \Delta^2 \phi^{(n+\frac{1}{2})} &= RG(\phi^{(n+\frac{1}{2})}) \text{ on } \Omega_1 \\ \left(\phi^{(n+\frac{1}{2})}, \frac{\partial \phi^{(n+\frac{1}{2})}}{\partial n} \right) &= \left(\phi^{(n)}, \frac{\partial \phi^{(n)}}{\partial n} \right) \text{ on } \partial\Omega_1 \end{aligned}$$

and

$$\begin{aligned} \Delta^2 \phi^{(n+1)} &= RG(\phi^{(n+1)}) \text{ on } \Omega_2 \\ \left(\phi^{(n+1)}, \frac{\partial \phi^{(n+1)}}{\partial n} \right) &= \left(\phi^{(n+\frac{1}{2})}, \frac{\partial \phi^{(n+\frac{1}{2})}}{\partial n} \right) \text{ on } \partial\Omega_2. \end{aligned}$$

The result is that this sequence converges in H^2 to the exact solution provided $R < C/(M + 1)$, where C is a constant depending on the geometry and is less than one, and $M = \|\phi^{(0)} - \phi\|_{H^2}$. We are now attempting to show a local convergence result for Reynolds numbers larger than one.

In this paper, we showed how Schwarz Alternating Methods can be imbedded within the framework of Banach and Schauder fixed point theories and Global Inversion theory to construct solutions of 2nd-order nonlinear elliptic PDEs. Future work include Schwarz methods for multiple subdomains, nonlinear parabolic and hyperbolic PDEs and the consideration of Schwarz methods on nonoverlapping subdomains.

References

1. A. Ambrosetti and G. Prodi, *A primer of nonlinear analysis*, Cambridge University Press, Cambridge, 1993.
2. J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition*, Math. Comput. **57** (1991), 1–21.
3. X. C. Cai and M. Dryja, *Domain decomposition methods for monotone nonlinear elliptic problems*, Domain decomposition methods in scientific and engineering computing (Providence, R.I.) (D. Keyes and J. Xu, eds.), AMS, 1994, pp. 335–360.
4. X. C. Cai, D. E. Keyes, and V. Venkatakrishnan, *Newton-Krylov-Schwarz: An implicit solver for CFD*, Proceedings of the Eight International Conference on Domain Decomposition Methods in Science and Engineering (New York) (R. Glowinski et al., ed.), Wiley, 1997.
5. T. F. Chan and T. P. Mathew, *Domain decomposition algorithms*, Acta Numerica (1994), 61–143.
6. M. Dryja and W. Hackbusch, *On the nonlinear domain decomposition method*, BIT (1997), 296–311.
7. P. L. Lions, *On the Schwarz alternating method I*, First Int. Symp. on Domain Decomposition Methods (Philadelphia) (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux, eds.), SIAM, 1988, pp. 1–42.
8. ———, *On the Schwarz alternating method II*, Second Int. Conference on Domain Decomposition Methods (Philadelphia) (T. F. Chan, R. Glowinski, J. Periaux, and O. Widlund, eds.), SIAM, 1989, pp. 47–70.
9. S. H. Lui, *On Schwarz alternating methods for nonlinear elliptic problems I*, Preprint (1997).
10. L. Nirenberg, *Topics in nonlinear functional analysis*, Courant Institute of Mathematical Sciences, New York University, New York, 1974.
11. B. F. Smith, P. Bjorstad, and W. D. Gropp, *Domain decomposition : Parallel multilevel algorithms for elliptic partial differential equations*, Cambridge University Press, New York, 1996.
12. X. C. Tai, *Domain decomposition for linear and nonlinear elliptic problems via function or space decomposition*, Domain decomposition methods in scientific and engineering computing (Providence, R.I.) (D. Keyes and J. Xu, eds.), AMS, 1994, pp. 335–360.
13. X. C. Tai and M. Espedal, *Rate of convergence of some space decomposition methods for linear and nonlinear problems*, SIAM J. Numer. Anal. (1998).
14. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances **1** (1994), 121–220.
15. J. Xu, *Two-grid discretization techniques for linear and nonlinear PDEs*, SIAM J. Numer. Anal. **33** (1996), 1759–1777.

Convergence Results for Non-Conforming hp Methods: The Mortar Finite Element Method

Padmanabhan Seshaiyer and Manil Suri

1. Introduction

In this paper, we present uniform convergence results for the mortar finite element method (which is an example of a non-conforming method), for h , p and hp discretizations over *general* meshes. Our numerical and theoretical results show that the mortar finite element method is a good candidate for hp implementation and also that the optimal rates afforded by the conforming h , p and hp discretizations are preserved when this non-conforming method is used, even over highly non-quasiuniform meshes.

Design over complex domains often requires the concatenation of separately constructed meshes over subdomains. In such cases it is difficult to coordinate the submeshes so that they conform over interfaces. Therefore, non-conforming elements such as the mortar finite element method [2, 3, 4] are used to “glue” these submeshes together. Such techniques are also useful in applications where the discretization needs to be selectively increased in localized regions (such as those around corners or other features) which contribute most to the pollution error in any problem. Moreover, different variational problems in different subdomains can also be combined using non-conforming methods.

When p and hp methods are being used, the interface incompatibility may be present not only in the meshes but also in the *degrees* chosen on the elements from the two sides. Hence the concatenating method used must be formulated to accomodate various degrees, and also be stable and optimal *both* in terms of mesh refinement (h version) *and* degree enhancement (p version). Moreover, this stability and optimality should be preserved when highly non-quasiuniform meshes are used around corners (such as the geometrical ones in the hp version).

We present theoretical convergence results for the mortar finite element method from [7],[8] and extend these in two ways in this paper. First, we show that the stability estimates established for the mortar projection operator (Theorem 2 in

1991 *Mathematics Subject Classification*. Primary 65N30; Secondary 65N15.

Key words and phrases. p Version, hp Version, Mortar Elements, Finite Elements, Non-Conforming.

This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command, USAF under Grant F49620-95-1-0230 and the National Science Foundation under Grant DMS-9706594.

[8]) are **optimal**. Second, we present h , p and hp computations for a Neumann problem, which fills a gap in numerical validation as explained in Section 4.

2. The Mortar Finite Element Method

We begin by defining the mortar finite element method for the following model problem.

$$(1) \quad -\Delta u = f, \quad u = 0 \text{ on } \partial\Omega_D, \quad \frac{\partial u}{\partial n} = g \text{ on } \partial\Omega_N.$$

where $\Omega \subset \mathbb{R}^2$ is a bounded polygonal domain with boundary $\partial\Omega = \overline{\partial\Omega_D} \cup \overline{\partial\Omega_N}$ ($\partial\Omega_D \cap \partial\Omega_N = \emptyset$), and for simplicity it is assumed $\partial\Omega_D \neq \emptyset$. Defining $H_D^1(\Omega) = \{u \in H^1(\Omega) | u = 0 \text{ on } \partial\Omega_D\}$ (we use Standard Sobolev space notation), we get the variational form of (1) : Find $u \in H_D^1(\Omega)$ satisfying, for all $v \in H_D^1(\Omega)$,

$$(2) \quad a(u, v) \stackrel{\text{def}}{=} \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} fv \, dx + \int_{\partial\Omega_N} gv \, ds \stackrel{\text{def}}{=} F(v).$$

This problem has a unique solution.

We now assume Ω is partitioned into non-overlapping polygonal subdomains $\{\Omega_i\}_{i=1}^K$ assumed to be geometrically conforming for simplicity (though our results also hold for the geometrically non-conforming (see [2]) case). The interface set Γ is defined to be the union of the interfaces $\Gamma_{ij} = \Gamma_{ji}$, i.e. $\Gamma = \cup_{i,j} \Gamma_{ij}$ where $\Gamma_{ij} = \partial\Omega_i \cup \partial\Omega_j$. Γ can then be decomposed into a set of disjoint straight line pieces $\gamma_i, i = 1, 2, \dots, L$. We denote $Z = \{\gamma_1, \dots, \gamma_L\}$.

Each Ω_i is assumed to be further subdivided into triangles and parallelograms by geometrically conforming, shape regular [5] families of meshes $\{\mathcal{T}_h^i\}$. The triangulations over different Ω_i are assumed independent of each other, with no compatibility enforced across interfaces. The meshes do not have to be quasiuniform and can be quite general, with only a mild restriction, Condition(M), imposed below.

For $K \subset \mathbb{R}^n$, let $\mathcal{P}_k(K)$ ($\mathcal{Q}_k(K)$) denote the set of polynomials of total degree (degree in each variable) $\leq k$ on K . We assume we are given families of piecewise polynomial spaces $\{V_{h,k}^i\}$ on the Ω_i ,

$$V_{h,k}^i = \{u \in H^1(\Omega_i) \mid u|_K \in \mathcal{S}_k(K) \text{ for } K \in \mathcal{T}_h^i, \quad u = 0 \text{ on } \partial\Omega_i \cap \partial\Omega_D\}.$$

Here $\mathcal{S}_k(K)$ is $\mathcal{P}_k(K)$ for K a triangle, and $\mathcal{Q}_k(K)$ for K a parallelogram. Note that $V_{h,k}^i$ are *conforming* on Ω_i , i.e. they contain continuous functions that vanish on $\partial\Omega_D$.

We define the space $\tilde{V}_{h,k}$ by,

$$(3) \quad \tilde{V}_{h,k} = \{u \in L_2(\Omega) \mid u|_{\Omega_i} \in V_{h,k}^i \quad \forall i\}$$

and a discrete norm over $\tilde{V}_{h,k} \cup H^1(\Omega)$ by,

$$(4) \quad \|u\|_{1,d}^2 = \sum_{i=1}^K \|u\|_{H^1(\Omega_i)}^2.$$

The condition on the mesh, which will be satisfied by almost any kind of mesh used in the h , p or hp version, is given below. Essentially, it says the refinement cannot be stronger than geometric.

Condition(M) *There exist constants α, C_0, ρ , independent of the mesh parameter h and degree k , such that for any trace mesh on $\gamma \in Z$, given by $x_0 <$*

$x_1 < \dots < x_{N+1}$, with $h_j = x_{j+1} - x_j$, we have $\frac{h_i}{h_j} \leq C_0 \alpha^{|i-j|}$ where α satisfies $1 \leq \alpha < \min\{(k+1)^2, \rho\}$.

To define the “mortaring”, let $\gamma \in Z$ be such that $\gamma \subset \Gamma_{ij}$. Since the meshes T_h^i are not assumed to conform across interfaces, two separate trace meshes can be defined on γ , one from Ω_i and the other from Ω_j . We assume that one of the indices i, j , say i , has been designated to be the *mortar index associated with γ* , $i = M(\gamma)$. The other is then the *non-mortar index*, $j = NM(\gamma)$. We then denote the trace meshes on γ by $T_{M(\gamma)}^h$ and $T_{NM(\gamma)}^h$, with the corresponding trace spaces being $V^M(\gamma)$ and $V^{NM}(\gamma)$, where e.g.

$$V^M(\gamma) = V_{h,k}^M(\gamma) = \{u|_\gamma \mid u \in V_{h,k}^i\}.$$

Given $u \in \tilde{V}_{h,k}$, we denote the mortar and non-mortar traces of u on γ by u_γ^M and u_γ^{NM} respectively. We now restrict the space $\tilde{V}_{h,k}$ by introducing constraints on the differences $u_\gamma^M - u_\gamma^{NM}$. This “mortaring” is accomplished via Lagrange Multiplier spaces $S(\gamma)$ defined on the non-mortar trace meshes $T_{NM(\gamma)}^h$. Let the subintervals of this mesh on γ be given by I_i , $0 \leq i \leq N$. Then we set $S(\gamma) = S_{h,k}^{NM}(\gamma)$ defined as,

$$S(\gamma) = \{\chi \in C(\gamma) \mid \chi|_{I_i} \in \mathcal{P}_k(I_i), i = 1, \dots, N-1, \chi|_{I_0} \in \mathcal{P}_{k-1}(I_0),$$

$$\chi|_{I_N} \in \mathcal{P}_{k-1}(I_N)\}$$

i.e. $S(\gamma)$ consists of piecewise continuous polynomials of degree $\leq k$ on the mesh $T_{NM(\gamma)}^h$ which are one degree less on the first and last subinterval.

We now define $V_{h,k} \subset \tilde{V}_{h,k}$ by,

$$(5) \quad V_{h,k} = \{u \in \tilde{V}_{h,k} \mid \int_\gamma (u_\gamma^M - u_\gamma^{NM}) \chi \, ds = 0 \quad \forall \chi \in S_{h,k}^{NM}(\gamma), \forall \gamma \in Z\}.$$

Then our discretization to (2) is defined by: Find $u_{h,k} \in V_{h,k}$ satisfying, for all $v \in V_{h,k}$,

$$(6) \quad a_{h,k}(u_{h,k}, v) \stackrel{\text{def}}{=} \sum_{i=1}^K \int_{\Omega_i} \nabla u_{h,k} \cdot \nabla v \, dx = F(v).$$

THEOREM 1. [3] Problem (6) has a unique solution.

3. Stability and Convergence Estimates

Let $V_0^{NM}(\gamma)$ denote functions in $V^{NM}(\gamma)$ vanishing at the end points of γ . The stability and convergence of the approximate problem depends on the properties of the projection operator $\Pi_\gamma : L_2(\gamma) \rightarrow V_0^{NM}(\gamma)$ defined as follows: For $u \in L_2(\gamma)$, $\gamma \in Z$, $\Pi_\gamma u = \Pi_{\gamma,h}^k u$ is a function in $V_0^{NM}(\gamma)$ that satisfies,

$$(7) \quad \int_\gamma (\Pi_\gamma^{h,k} u) \chi \, ds = \int_\gamma u \chi \, ds \quad \forall \chi \in S_{h,k}^{NM}(\gamma).$$

Condition(M) imposed in the previous section is sufficient, as shown in [7, 8], to ensure the following stability result for the projections Π_γ .

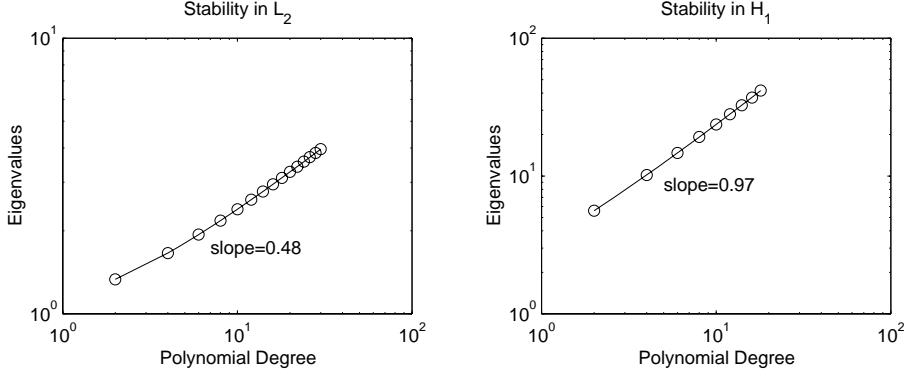


FIGURE 1. (a) Maximum eigenvalue for L_2 (b) Maximum eigenvalue for H^1

THEOREM 2. Let $\{V_{h,k}\}$ be such that Condition(M) holds. Let $\{\Pi_\gamma^{h,k}, \gamma \in Z\}$ be defined by (7). Then there exists a constant C , independent of h, k (but depending on α, C_0, ρ) such that,

$$(8) \quad \|\Pi_\gamma^{h,k} u\|_{0,\gamma} \leq Ck^{\frac{1}{2}} \|u\|_{0,\gamma} \quad \forall u \in L_2(\gamma)$$

$$(9) \quad \|(\Pi_\gamma^{h,k} u)'\|_{0,\gamma} \leq Ck \|u'\|_{0,\gamma} \quad \forall u \in H_0^1(\gamma)$$

A question unanswered in [8] was whether (8)–(9) are optimal. Figure 1 shows that the powers of k in (8)–(9) cannot be improved. This is done by approximating the norms of the operator $\|\Pi_\gamma^{h,k}\|_{\mathcal{L}(L_2(\gamma), L_2(\gamma))}$ and $\|\Pi_\gamma^{h,k}\|_{\mathcal{L}(H^1(\gamma), H^1(\gamma))}$ (with h fixed), using an eigenvalue analysis. (For details we refer to the thesis [7].) It is observed that these norms grow as $O(k^{\frac{1}{2}})$ and $O(k)$ respectively, as predicted by Theorem 2.

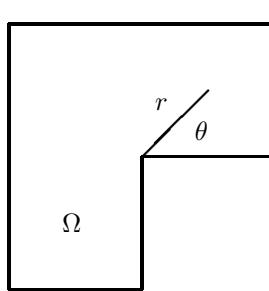
Using Theorem 2 and an extension result for hp meshes [8], we can prove our main theorem, by the argument used in [3], Theorem 2 (see [7, 8] for details). In the theorem below, $\{N_j\}$ denotes the set of all end points of the segments $\gamma \in Z$.

THEOREM 3. Let $\{V_{h,k}\}$ be such that Condition(M) holds. Then for any $\epsilon > 0$, there exists a constant $C = C(\epsilon)$, independent of u, h and k such that,

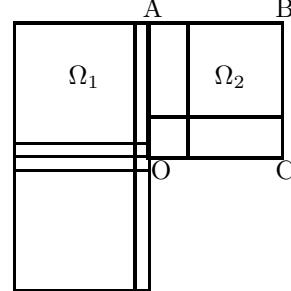
$$(10) \quad \|u - u_{h,k}\|_{1,d} \leq C \sum_{\gamma \in Z} \inf_{\psi \in S_{h,k}(\gamma)} \left\| \frac{\partial u}{\partial n} - \psi \right\|_{(H^{\frac{1}{2}}(\gamma))'} + \\ C \inf_{\substack{v \in \tilde{V}_{h,k} \\ v(N_j) = u(N_j)}} \left\{ \sum_i \|u - v\|_{1,\Omega_i} + \right. \\ \left. k^{\frac{3}{4}+\epsilon} \sum_{\gamma \in Z} \left(\|u - v_\gamma^M\|_{\frac{1}{2}+\epsilon,\gamma} + \|u - v_\gamma^{NM}\|_{\frac{1}{2}+\epsilon,\gamma} \right) \right\}$$

Moreover, for h or k fixed, or for quasiuniform meshes, we may take $\epsilon = 0$ if we replace $\|\cdot\|_{\frac{1}{2}+\epsilon,\gamma}$ by $\|\cdot\|_{H_{00}^{\frac{1}{2}}}(\gamma)$.

The following estimate for quasiuniform meshes follows readily from Theorem 3:



Fig(a)



Fig(b)

FIGURE 2. (a) L-shaped domain (b) Tensor product mesh for $m = n = 2$

THEOREM 4. *Let the solution u of (2) satisfy $u \in H^l(\Omega)$, $l > \frac{3}{2}$ ($l > \frac{7}{4}$ if k varies). For the hp version with quasiuniform meshes $\{\mathcal{T}_h^i\}$ on each Ω_i ,*

$$(11) \quad \|u - u_{h,k}\|_{1,d} \leq Ch^{\mu-1}k^{-(l-1)+\frac{3}{4}}\|u\|_{l,\Omega}$$

where $\mu = \min\{l, k+1\}$ and C is a constant independent of h, k and u .

Theorem 3 also tells us that, using highly non-quasiuniform *radical* meshes in the neighbourhood of singularities (see Section 4 of [1]), we can now recover full $O(h^k)$ convergence even when the mortar element method is used. Moreover, exponential convergence that is realized when the (conforming) hp version is used over *geometrical* meshes will be preserved when the non-conforming mortar finite element is used. We illustrate these results computationally in the next section.

4. Numerical Results

We consider problem (1) on the L-shaped domain shown in Figure 2, which is partitioned into two rectangular subdomains, Ω_1 and Ω_2 , by the interface AO. In [8], we only considered the case where $\partial\Omega_D = \partial\Omega$. This, however, results in the more restrictive mortar method originally proposed in [3], where continuity is enforced at vertices of Ω_i . To implement the method proposed in [2] and analyzed here, where the vertex continuity enforcement is removed, we must take Neumann conditions at the ends of AO. We therefore consider here the Neumann case where $\partial\Omega_N = \partial\Omega$, with uniqueness maintained by imposing the condition $u = 0$ at the single point C. Our exact solution is given by,

$$u(r, \theta) = r^{\frac{2}{3}} \cos\left(\frac{2\theta}{3}\right) - 1.$$

where (r, θ) are polar coordinates with origin at O. We use the mixed method to implement the mortar condition. For our computations, we consider tensor product meshes where Ω_2 is divided into n^2 rectangles and Ω_1 is divided into $2m^2$ rectangles (see Figure 2).

It is well-known that this domain will result in a strong $r^{\frac{2}{3}}$ singularity which occurs at the corner O in Figure 2, which limits the convergence to $O(N^{-\frac{1}{3}})$ when the quasiuniform h version is used. Figure 3 shows that this rate is preserved when the mortar finite element is used (graph (1)) with degree $k = 2$ elements. When

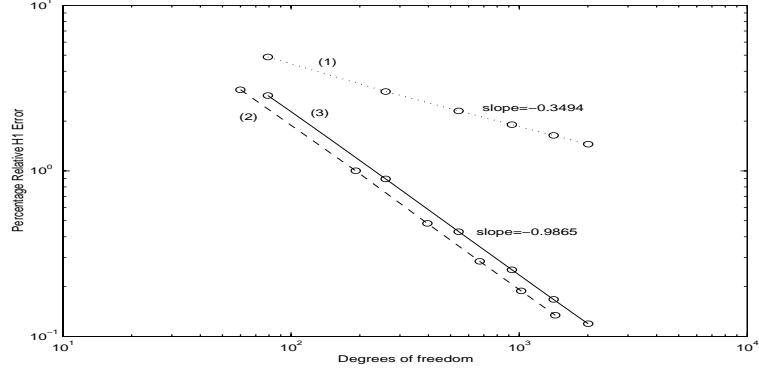


FIGURE 3. The relative error in the energy norm in dependence on h for radical meshes ($k = 2$)

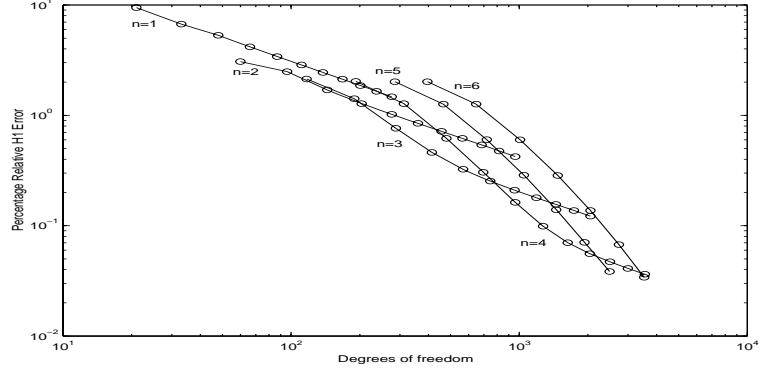


FIGURE 4. The relative error in the energy norm in dependence on N for geometric meshes ($\sigma_1 = 0.17$, $\sigma_2 = 0.13$)

suitably refined radical meshes are used, then $O(N^{-1})$ convergence is recovered both for the conforming (graph(2)) and mortar (graph(3)) methods.

For the p and hp mortar FEM on geometric meshes, we take $m = n$ and consider the geometric ratio σ (i.e. the ratio of the sides of successive elements, see [6]) to vary in each domain Ω_i . The optimal value is 0.15 (see [6]), but we take $\sigma_1 = 0.17$ and $\sigma_2 = 0.13$ to make the method non-conforming. We observe in Figure 4, the typical p convergence for increasing degree k for various n . Note that for our problem, at least, we do not see the loss of $O(k^{\frac{2}{3}})$ in the asymptotic rate due to the projection Π_γ not being completely stable (as predicted by Theorem 4 and Figure 1). See Figure 5(a) where we have plotted the case $\sigma_1 = 0.17, \sigma_2 = 0.13$ for $n = 4$ together with the conforming cases $\sigma_1 = \sigma_2 = 0.13$ and 0.17. The results indicate that the p version mortar FEM behaves almost identically to the conforming FEM.

Finally, in Figure 5(b), we plot $\log(\text{relative error})$ vs $N^{\frac{1}{4}}$, which gives a straight line, showing the exponential rate of convergence. We also plot $\log(\text{relative error})$ vs $N^{\frac{1}{3}}$, which is the theoretical convergence rate for the *optimal* geometric mesh (see [1], [6]). Since we consider a tensor product mesh here, which contains extra

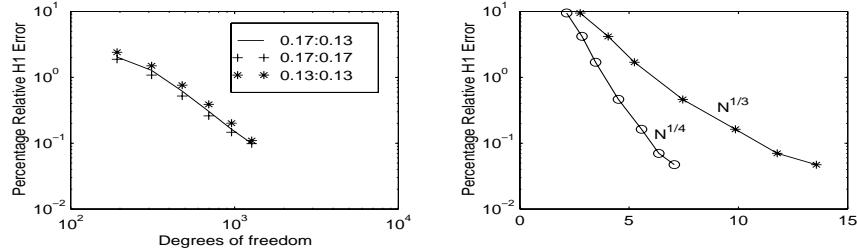


FIGURE 5. (a) Performance of the mortar FEM for $n=4$ (b) Exponential Convergence for the hp mortar FEM

degrees of freedom, we can only obtain an exponential convergence rate of $Ce^{-\gamma N^{\frac{1}{4}}}$ theoretically.

References

1. I. Babuška and M. Suri, *The p and $h-p$ versions of the finite element method: basic principles and properties*, SIAM Review **36** (1994), 578–632.
2. F. Ben Belgacem, *The mortar finite element method with Lagrange multipliers*, to appear in Numer. Math., 1999.
3. C. Bernardi, Y. Maday, and A. T. Patera, *Domain decomposition by the mortar element method*, Asymptotic and Numerical Methods for PDEs with Critical Parameters (1993), 269–286.
4. M. Casarin and O. B. Widlund, *A hierarchical preconditioner for the mortar finite element method*, ETNA, Electron. Trans. Numer. Anal. **4** (1996), 75–88.
5. P. G. Ciarlet, *The finite element method for elliptic problems*, North Holland, Amsterdam, 1978.
6. W. Gui and I. Babuška, *The hp version of the finite element method in one dimension*, Numer. Math. **40** (1986), 577–657.
7. P. Seshaiyer, *Non-conforming hp finite element methods*, Ph.D. Dissertation, University of Maryland Baltimore County, 1998 (expected).
8. P. Seshaiyer and M. Suri, *Uniform hp convergence results for the mortar finite element method*, Submitted to Math. Comp., 1997.

DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF MARYLAND BALTIMORE COUNTY, BALTIMORE, MD 21250, USA.

E-mail address: padhu@math.umbc.edu

DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF MARYLAND BALTIMORE COUNTY, BALTIMORE, MD 21250, USA.

E-mail address: suri@math.umbc.edu

Intergrid Transfer Operators for Biharmonic Problems Using Nonconforming Plate Elements on Nonnested Meshes

Zhongci Shi and Zhenghui Xie

1. Introduction

The aim of this paper is to construct a preconditioner for biharmonic problems using nonconforming plate element on nonnested meshes by additive Schwarz methods. The success of the methods depends heavily on the existence of a uniformly or nearly uniformly bounded decomposition of a function space in which the problem is defined, and intergrid transfer operators with certain stable approximation properties play an important role in the decomposition [1, 2, 4, 5, 7, 8, 3]. For the case when coarse and fine spaces are all nonconforming, a natural intergrid operator seems to be one defined by taking averages of the nodal parameters. We define an intergrid transfer operator for nonconforming plate elements in this natural way, discuss its stable approximation properties, and obtain the stable factor $(H/h)^{3/2}$. It is also shown that the stable factor cannot be improved. However, to get an optimal preconditioner, we need in general the stability with a factor C independent of mesh parameters H and h . Therefore, it cannot be used for that purpose. To obtain an optimal preconditioner for biharmonic problems using nonconforming plate elements on nonnested meshes by additive Schwarz methods, we define an intergrid transfer operator, prove certain stable approximation properties, construct a uniformly bounded decomposition for the finite element space, and then get optimal convergence properties with a not necessarily shape regular subdomain partitioning. Here the fine mesh may not be quasi-uniform.

2. A sharp estimate

Let Ω be a bounded polygonal domain in R^2 with boundary $\partial\Omega$. We consider the following biharmonic Dirichlet problem:

$$(1) \quad \Delta^2 u = f \text{ in } \Omega, u = \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega.$$

1991 *Mathematics Subject Classification*. Primary 65F10; Secondary 65N30, 65N55.

Key words and phrases. Intergrid Transfer Operator, Additive Schwarz Method, Biharmonic Equation, Nonconforming Plate Elements, Nonnested Meshes.

This work was supported by Chinese National Key Project of Fundamental Research: Methods and Theories in Large-scale Scientific and Engineering Computing.

The variational form of the problem (1) is : to find $u \in H_0^2(\Omega)$ such that

$$(2) \quad a(u, v) = (f, v), \forall v \in H_0^2(\Omega),$$

where

$$\begin{aligned} a(u, v) &= \int_{\Omega} \{\Delta u \Delta v + (1 - \sigma)(2\partial_{12}u\partial_{12}v - \partial_{11}u\partial_{22}v - \partial_{22}u\partial_{11}v)\} dx, \\ (f, v) &= \int_{\Omega} fv dx, \sigma \in (0, 0.5) \text{ is the Poisson ratio .} \end{aligned}$$

The unique solvability of the problem (2) for $f \in L^2(\Omega)$ follows from the continuity and coerciveness of the bilinear form in $H_0^2(\Omega)$ and Lax-Milgram theorem.

Let J_h be a triangulation of Ω , and V_h a nonconforming plate element space. The corresponding finite element discrete equation for problem (2) is : to find $u_h \in V_h$ such that

$$(3) \quad a_h(u_h, v_h) = (f, v_h), \quad \forall v \in V_h,$$

where

$$a_h(u_h, v_h) = \sum_{\tau \in J_h} \int_{\tau} \{\Delta u_h \Delta v_h + (1 - \sigma)(2\partial_{12}u_h\partial_{12}v_h - \partial_{11}u_h\partial_{22}v_h - \partial_{22}u_h\partial_{11}v_h)\} dx.$$

To obtain an optimal preconditioner of problem(3) by additive Schwarz methods, the intergrid transfer operator with certain stable approximation properties plays an important role. For the case when coarse and fine spaces are all nonconforming, a natural intergrid transfer operator seems to be one defined by taking the average of the values of the nodal parameters. We define an intergrid transfer operator in this natural way. However, it can be shown that the intergrid transfer operator is not suitable for obtaining an optimal preconditioner.

We take Morley element as an example. In this section, let J_H and J_h be two quasi-uniform triangulations of Ω and V_H the associated Morley element spaces. We assume that J_h is a refinement of J_H . Note that $V_H \not\subset V_h$.

The intergrid transfer operator $I_H^h : V_H \rightarrow V_h$ is defined as follows.

For $v \in V_H$, $I_H^h v \in V_h$ is defined so that

a) if p is a vertex of J_h which is also a vertex of J_H or in the interior of $\tau \in J_H$, $(I_H^h v)(p) = v(p)$; for other vertices p of J_h , v may have a jump at p and $I_H^h v$ takes the averages of v at p ;

b) if m is a midpoint of an edge of J_h which is in the interior of $\tau \in J_H$, $\frac{\partial(I_H^h v)}{\partial n}(m) = \frac{\partial v}{\partial n}(m)$; for one of other edge midpoints m associated with J_h , $\frac{\partial v}{\partial n}$ may have several jumps and $\frac{\partial(I_H^h v)}{\partial n}(m)$ takes the arithmetic average value of $\frac{\partial v}{\partial n}$ at m .

About the operator I_H^h we have the following sharp estimates.

THEOREM 1. For $v \in V_H$, we have

$$(4) \quad |I_H^h v - v|_{0,h,\Omega} \leq C(H^3 h)^{1/2} |v|_{2,H,\Omega},$$

$$(5) \quad |I_H^h v - v|_{1,h,\Omega} \leq C \left(\frac{H^3}{h} \right)^{1/2} |v|_{2,H,\Omega},$$

and

$$(6) \quad |I_H^h v|_{2,h,\Omega} \leq C \left(\frac{H}{h} \right)^{3/2} |v|_{2,H,\Omega}.$$

Furthermore, the estimates (5) and (6) are sharp.

The proof of the theorem can be found in Shi and Xie [6]. For other nonconforming plate elements, we can get similar results.

To get an optimal preconditioner, we need in general the stability with a factor C independent of mesh parameters H and h . Therefore, it cannot be used for obtaining an optimal preconditioner.

3. An intergrid transfer operator for nonconforming plate element on nonnested meshes

We now define another intergrid transfer operator, discuss its stable approximation properties, and construct an optimal preconditioner for problem(2) using nonconforming plate elements on nonnested meshes.

3.1. Stable approximation properties. Let J_{H_c} be a quasi-uniform triangulation of Ω . J_{H_c} will be referred to as the coarse grid. Here H_c is the maximum diameter of this coarse triangulation. Let J_h be a triangulation of Ω that satisfies the minimal angle condition in this section. In general, J_h is not a subdivision of J_{H_c} . We assume that each fine triangle intersects with at most n_0 coarse triangles, $n_0 \leq C$,

$$(7) \quad \begin{cases} h \leq CH_c, \text{ and} \\ |\tau| \leq C|k|, \text{ if } \bar{k} \cap \bar{\tau} \neq \emptyset, \tau \in J_h, k \in J_{H_c}, \end{cases}$$

where $|\cdot|$ means the area in R^2 .

Let V_{H_c} be Morley element spaces associated with meshes H_c , nodal parameters of which vanish on $\partial\Omega$. Note that $V_{H_c} \not\subset V_h$. For the space V_{H_c} , we take $W_{H_c} = AR^{H_c}(\Omega)$ to be its conforming relative, where W_{H_c} is the P_5 Argyris element space $\{w \in C^1(\bar{\Omega}) : w|_T \in P_5(T), \forall T \in J_{H_c}, w = \partial_n w = 0 \text{ on } \partial\Omega\}$. The conforming interpolation operator $E_{H_c} : V_{H_c} \rightarrow W_{H_c}$ is defined as follows(cf. Brenner [1]):

$$(8) \quad \begin{cases} (E_{H_c}v)(p) = v(p); \\ (D^\alpha E_{H_c}v)(p) = \text{average of } (D^\alpha v_i)(p), |\alpha| = 1; \\ D^\alpha E_{H_c}v(p) = 0, |\alpha| = 2; \\ \partial_n E_{H_c}v(m) = \partial_n v(m); \end{cases}$$

where p is vertex, m is midpoint of sides, $v_i = v|_{T_i}$ and T_i contains p as a vertex. $E_h : V_h \rightarrow W_h$ can be defined similarly. We have [1]

$$(9) \quad \|v - E_{H_c}v\|_{L^2(T)} + H_c|v - E_{H_c}v|_{1,T} + H_c^2|E_{H_c}v|_{2,T} \leq CH_c^2|v|_{2,T}, \forall v \in V_{H_c},$$

where $T \in J_{H_c}$, and

$$(10) \quad \|w - E_h w\|_{L^2(\tau)} + h_\tau|w - E_h w|_{1,\tau} + h_\tau^2|E_h w|_{2,\tau} \leq Ch_\tau^2|w|_{2,\tau}, \forall w \in V_h,$$

where $\tau \in J_h$, h_τ is the diameter of τ .

Define the nodal interpolation operator $\Pi_{H_c} : C_0^1(\Omega) \rightarrow V_{H_c}$ as follows:

$$\begin{cases} \Pi_{H_c}v(p) = v(p), \\ \partial_n \Pi_{H_c}v(m) = \partial_n v(m). \end{cases}$$

$\Pi_h : C_0^1(\Omega) \rightarrow V_h$ can be defined similarly. Then, it is easy to prove that

(11)

$$\|v - \Pi_{H_c}v\|_{L^2(T)} + H_c|v - \Pi_{H_c}v|_{1,T} + H_c^2|\Pi_{H_c}v|_{2,T} \leq CH_c^2|v|_{2,T}, \forall v \in H^2(T),$$

where $T \in J_{H_c}$.

The intergrid transfer operator $I_{H_c}^h : V_{H_c} \rightarrow V_h$ is defined by $I_{H_c}^h = \Pi_h \cdot E_{H_c}$. For nested meshes, certain stable approximation properties of the intergrid transfer operator were discussed in Brenner [1]. However, for nonnested meshes, it can not be proved in the same way. We have the following theorem, which plays an important role in our analysis.

THEOREM 2. *There exists a constant $C > 0$, independent of h, H_c such that for $u \in V_{H_c}$,*

$$(12) \quad |I_{H_c}^h u|_{2,h,\Omega} \leq C |u|_{2,H_c,\Omega},$$

$$(13) \quad \|u - I_{H_c}^h u\|_{0,\Omega} + H_c |E_{H_c} u - I_{H_c}^h u|_{1,h,\Omega} \leq CH_c^2 |u|_{2,H_c,\Omega},$$

where $|u|_{i,H_c,\Omega}^2 = \sum_{T \in J_{H_c}} |u|_{i,T}^2$.

PROOF. We first prove (12). Let $\bar{u} = E_{H_c} u$. The essential step is to establish the estimate

$$(14) \quad |I_{H_c}^h u|_{H^2(k)}^2 \leq C \sum_{\tau \cap \bar{k} \neq \emptyset, \tau \in J_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 |k|, \quad \forall u \in V_{H_c}, \text{ here } k \in J_h.$$

Let $\tau = \Delta p_1 p_2 p_3$, and m_1, m_2, m_3 be the midpoint of the edge $\overline{p_2 p_3}$, $\overline{p_3 p_1}$, and $\overline{p_1 p_2}$ of τ , respectively. If k belongs completely to a single coarse element τ , $\tau \in J_{H_c}$, then (14) is obviously true. We now prove (14) in the case that k does not belong completely to arbitrary coarse element τ , $\tau \in J_{H_c}$. We know that

$$(15) \quad |I_{H_c}^h u|_{H^2(k)}^2 \leq C \sum_{i=1}^3 (\partial_n(\bar{u} - \bar{u}_I)(m_i))^2,$$

where \bar{u}_I means the linear interpolation of \bar{u} . Let $\overline{p_2 m_1}$ be the line segment connecting points p_2 and m_1 . We assume that $\overline{p_2 m_1}$ is cut into l pieces by the coarse triangles $\tau_1^{p_2 m_1}, \dots, \tau_l^{p_2 m_1}$, and $u(\cdot)$ is a polynomial on each piece. By the assumption made at the beginning of this section, $l \leq C$. Therefore, by using the triangle inequality, we have

$$(16) \quad |\partial_n(\bar{u} - \bar{u}_I)(m_1)|^2 \leq 2|\partial_n(\bar{u} - \bar{u}_I)(p_2)|^2 + 2|\partial_n \bar{u}(p_2) - \partial_n \bar{u}(m_1)|^2 \equiv I + II.$$

Let $g = \bar{u} - \bar{u}_I$, then $g(p_1) = g(p_2) = g(p_3) = 0$, and there exists $\xi_1 \in \overline{p_1 p_2}, \xi_2 \in \overline{p_2 p_3}$ such that

$$(17) \quad \partial_{\overline{p_1 p_2}} g(\xi_1) = 0, \partial_{\overline{p_2 p_3}} g(\xi_2) = 0.$$

Hence using the triangle inequality and the mean value theorem, we obtain

$$(18) \quad |\partial_{\overline{p_1 p_2}} g(p_2)|^2 = |\partial_{\overline{p_1 p_2}} g(p_2) - \partial_{\overline{p_1 p_2}} g(\xi_1)|^2 \leq C \sum_{\tau \cap \bar{k} \neq \emptyset, \tau \in J_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 |k|,$$

and

$$(19) \quad |\partial_{\overline{p_2 p_3}} g(p_2)|^2 = |\partial_{\overline{p_2 p_3}} g(p_2) - \partial_{\overline{p_2 p_3}} g(\xi_2)|^2 \leq C \sum_{\tau \cap \bar{k} \neq \emptyset, \tau \in J_{H_c}} |u|_{2,\infty,\tau}^2 |k|.$$

From (16), (18)-(19) we have

$$(20) \quad I \leq C \sum_{\tau \cap \bar{k} \neq \emptyset, \tau \in J_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 |k|.$$

Similarly,

(21)

$$II = 2|\partial_n \bar{u}(p_2) - \partial_n \bar{u}(m_1)| \leq C \sum_{m=1}^l |\bar{u}|_{2,\infty,\tau_m^{p_2 m_1}}^2 |k|, \leq C \sum_{\bar{\tau} \cap \bar{k} \neq \phi, \tau \in J_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 |k|.$$

For m_2, m_3 , we can get the estimates similar to (16), (20) and (21). Therefore, from (16), (20) and (21) and their similar estimates for m_2 and m_3 , we have

$$(22) \quad \sum_{i=1}^3 |\partial_n(\bar{u} - \bar{u}_I)(m_i)|^2 \leq C \sum_{\bar{\tau} \cap \bar{k} \neq \phi, \tau \in J_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 |k|.$$

(14) follows from (15) and (22).

For $\tau \in J_{H_c}$, we denote by $\tau_j, j = 1, \dots, l_1$, all the coarse triangles which share at least one of the fine triangles that intersects with τ (i.e., this fine triangle intersects with both τ and τ_j). $l_1 \leq C$.

For all $k_i \in J_h, i = 1, \dots, m'$, whose intersection with τ is nonempty, we obtain from (14) that

$$(23) \quad |I_{H_c}^h u|_{H^2(k_i)}^2 \leq C \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 |k_i|, \forall u \in V_{H_c}.$$

By summing (23) over all $k_i, i = 1, \dots, m'$, and from an elementwise inverse estimate we have

$$(24) \quad \begin{aligned} \sum_{k \cap \tau \neq \phi} |I_{H_c}^h u|_{H^2(k)}^2 &\leq C \sum_{i=1}^{m'} \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 |k_i| \leq C \sum_{j=1}^{l_1} |u|_{2,\infty,\tau_j}^2 \sum_{i=1}^{m'} |k_i| \\ &\leq C \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 |\tau| \leq C \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2. \end{aligned}$$

Here we used the fact that, for each τ , the sum of the areas of the fine triangle that intersects with τ is less than $C|\tau|$ because of the assumption $h \leq CH_c$.

By summing (24) over all τ in J_{H_c} and noting that the number of repetitions, for each τ , in the summation is finite, we have

$$(25) \quad |I_{H_c}^h u|_{2,h,\Omega}^2 \leq \sum_{\tau \in J_{H_c}} \sum_{\bar{k} \cap \bar{\tau} \neq \phi} |I_{H_c}^h u|_{H^2(k)}^2 \leq C |\bar{u}|_{2,\Omega}^2.$$

(12) follows from (25) and (9).

We now turn to the proof of (13). Let $k \in J_h$ be a fine triangle and p one of its nodes, which implies that $w(p) = 0, \partial_n w(m) = 0$. Here $w = u - I_{H_c}^h u$. We consider the integral

$$(26) \quad \begin{aligned} \|w\|_{L^2(k)}^2 &= \int_k w^2(x) dx \leq 2 \int_k (w(p) - w(x) - Dw(p).(x-p))^2 dx \\ &\quad + 2 \int_k |Dw(p).(x-p)|^2 dx \equiv I_1 + I_2. \end{aligned}$$

Let \overline{xp} be the line segment connecting points x and p . We assume \overline{xp} is cut into l_2 pieces by coarse triangles $\tau_1^k, \dots, \tau_{l_2}^k$. Using the triangle inequality and the mean

value theorem, we have

$$(27) \quad I_1 \leq 2 \int_k (w(p) - w(x) - Dw(p) \cdot (p - x))^2 dx \leq 2l_2 \sum_{m=1}^{l_2} |w|_{2,\infty,\tau_m^k \cap k}^2 h_k^4 |k|,$$

where h_k is the diameter of element k . For the three edge midpoints $m_i (i = 1, 2, 3)$ of k , and by using $\partial_n w(m_i) = 0$ and the arguments similar to (17)-(19) we

$$(28) \quad I_2 \leq 2l_2 \sum_{m=1}^{l_2} |w|_{2,\infty,\tau_m^k \cap k}^2 h_k^4 |k|.$$

It follows from (26)-(28) and an elementwise inverse estimate that

$$\begin{aligned} \|w\|_{L^2(k)}^2 &\leq C \sum_{\bar{\tau} \cap k \neq \phi, \tau \in V_{H_c}} |w|_{2,\infty,\tau \cap k}^2 h_k^4 |k| \\ (29) \quad &\leq C \sum_{\bar{\tau} \cap k \neq \phi, \tau \in V_{H_c}} (|\bar{u}|_{2,\infty,\tau}^2 + |I_{H_c}^h u|_{2,\infty,k}^2) h_k^4 |k| \\ &\leq C \sum_{\bar{\tau} \cap k \neq \phi, \tau \in V_{H_c}} |\bar{u}|_{2,\infty,\tau}^2 h_k^4 |k| + Cn_0 h^4 |\bar{u}|_{2,k}^2. \end{aligned}$$

For $\tau \in J_{H_c}$, from (29) and by the same notation as (23) we have

$$(30) \quad \|w\|_{L^2(k_i)}^2 \leq C \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 h^4 |k_i| + C |\bar{u}|_{2,k_i}^2 h^4.$$

By summing (30) over all $k_i, i = 1, \dots, m$, and using the argument similar to (24) and the fact that $\sum_{k \cap \tau \neq \phi} |k| \leq C|\tau|$ we have

$$\begin{aligned} (31) \quad &\sum_{\bar{k} \cap \bar{\tau} \neq \phi, k \in J_h} \|w\|_{L^2(k)}^2 = \sum_{i=1}^{m'} \|w\|_{L^2(k_i)}^2 \leq C \sum_{i=1}^{m'} \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 h^4 |k_i| + C \sum_{i=1}^{m'} |\bar{u}|_{2,k_i}^2 h^4 \\ &\leq Ch^4 \sum_{j=1}^{l_1} |\bar{u}|_{2,\infty,\tau_j}^2 \sum_{i=1}^{m'} |k_i| + Ch^4 \sum_{j=1}^{l_1} |\bar{u}|_{2,\tau_j}^2 \leq Ch^4 \sum_{j=1}^{l_1} |\bar{u}|_{2,\tau_j}^2. \end{aligned}$$

By summing (31) over τ in J_{H_c} and noting that the number of repetitions, for each τ , in the summation is finite, we obtain

$$(32) \quad \|w\|_{L^2(\Omega)}^2 \leq Ch^4 |\bar{u}|_{H^2(\Omega)}^2 \leq Ch^4 |u|_{H^2(\Omega)}^2,$$

and by the similar argument we can get

$$(33) \quad |w|_{1,h,\Omega}^2 \leq |E_{H_c} u - I_{H_c}^h u|_{1,h,\Omega}^2 \leq CH_c^2 |E_{H_c} u|_{2,\Omega}^2 \leq CH_c^2 |u|_{2,\Omega}^2.$$

Combining (32) and (33) completes the proof of the lemma. \square

We now partition Ω into nonoverlapping subdomains $\{\Omega_i\}$, such that no $\partial\Omega_i$ cuts through any elements $\tau, \tau \in J_h$, and $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$. Note that we do not assume that $\{\Omega_i\}$ forms a regular finite element subdivision of Ω , nor that the diameters of Ω_i are of the same order. To obtain an overlapping decomposition of Ω , we extend each Ω_i to a larger subdomain $\Omega'_i \supset \Omega_i$, which is also assumed not to cut any fine mesh triangles, such that $\text{dist}(\partial\Omega'_i \cap \Omega, \partial\Omega_i \cap \Omega) \geq C\delta, \forall i$, for a constant $C > 0$. Here $\delta > 0$ will be referred to as the overlapping size. We assume that there exists an integer N_c independent of the mesh parameters h, H_c and δ such that any point

in Ω can belong to at most N_c subdomains Ω'_i . For each $\Omega'_i, i = 1, \dots, N$, we define a finite element space

$$V_i = \{v \in V_h; \text{ nodal parameters } = 0 \text{ at } \partial\Omega'_i \text{ and outside } \Omega'_i\}.$$

On the basis of Theorem 2 and (9)-(11), we can prove the next theorem, which shows that the decomposition $V_h = V_0 + V_1 + \dots + V_N$ exists and is uniformly bounded when $\delta = O(H_c)$.

THEOREM 3. *For any $v \in V_h$, there exist $v_0 \in V_{H_c}, v_i \in V_i, i = 1, \dots, N$, such that*

$$v = I_{H_c}^h v_0 + v_1 + \dots + v_N,$$

and in addition, there exists a constant $C_0 > 0$, independent of the mesh parameters h, H_c and δ such that

$$a_{H_c}(v_0, v_0) + \sum_{i=1}^N a_h(v_i, v_i) \leq C_0 N_c \left(1 + \frac{H_c^2}{\delta^2} + \frac{H_c^4}{\delta^4} \right) a_h(v, v), \forall v \in V_h.$$

The proof can be found in Xie [7].

3.2. An Additive Schwarz Method. Define $A_h : V_h \rightarrow V_h, A_i : V_i \rightarrow V_i (1 \leq i \leq N)$, and $A_{H_c} : V_{H_c} \rightarrow V_{H_c}$ by

$$\begin{aligned} (A_h v, w) &= a_h(v, w), \forall v, w \in V_h, \\ (A_i v, w) &= a_h(v, w), \forall v, w \in V_i, \\ (A_{H_c} v, w) &= a_{H_c}(v, w), \forall v, w \in V_{H_c}, \end{aligned}$$

respectively. The operator $Q_i : V_h \rightarrow V_i, 1 \leq i \leq N$, is defined by

$$(Q_i v, w) = (v, w), \forall v \in V_h, w \in V_i.$$

The operator $P_i : V_h \rightarrow V_i, 1 \leq i \leq N$, is defined by

$$a_h(P_i v, w) = a_h(v, w), \forall v \in V_h, w \in V_i.$$

The operators $I_h^{H_c}, P_h^{H_c} : V_h \rightarrow V_{H_c}$ are defined by

$$(I_{H_c}^h v, w) = (v, I_h^{H_c} w), \forall v \in V_{H_c}, w \in V_h,$$

and

$$a(I_{H_c}^h v, w) = a_h(v, P_h^{H_c} w), \forall v \in V_{H_c}, w \in V_h,$$

respectively.

The two level additive Schwarz preconditioner $B : V_h \rightarrow V_h$ is defined by

$$B := I_{H_c}^h A_{H_c}^{-1} I_h^{H_c} + \sum_{i=1}^N A_i^{-1} Q_i.$$

It can be easily seen that the operator $P = BA_h = I_{H_c}^h P_h^{H_c} + \sum_{i=1}^N P_i$ is symmetric positive-definite with respect to $a_h(\cdot, \cdot)$.

On the basis of Theorem 2 and Theorem 3, we obtain the following theorem which shows that P is uniformly bounded from both above and below when $\delta = O(H_c)$.

THEOREM 4. *The following estimate holds:*

$$\lambda_1 a_h(u, u) \leq a_h(Pu, u) \leq \lambda_2 a_h(u, u), \forall u \in V_h,$$

where

$$\lambda_2/\lambda_1 \leq CN_c \left(1 + \frac{H_c^2}{\delta^2} + \frac{H_c^4}{\delta^4} \right),$$

which is independent of the diameter of subdomains. This allows us to use subdomains of arbitrary shape.

Acknowledgment. The second author would like to thank Professors Lieheng Wang, Dehao Yu and Jingchao Xu for valuable discussions.

References

1. S.C. Brenner, *A two-level additive Schwarz preconditioner for nonconforming plate elements*, Proceedings of DDM7: Domain Decomposition Methods in Scientific and Engineering Computing (J. Xu D. Keyes, ed.), 1994, pp. 9–14.
2. X.C. Cai, *The use of pointwise interpolation in domain decomposition methods with non-nested methods*, SIAM J.Sci.Comput. **16** (1995), 250–256.
3. T.F. Chan, B.F. Smith, and J. Zou, *Overlapping Schwarz method on unstructured meshes using non-matching grids*, Numer.Math. **73** (1996), 149–167.
4. M. Dryja and O.B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. (1994), 604–620.
5. M. Sarkis, *Two-level Schwarz methods for nonconforming finite elements and discontinuous coefficients*, Tech. Report 629, Department of Computer Science, Courant Institute, 1993.
6. Z.C. Shi and Z.H. Xie, *Sharp estimates on intergrid transfer operators for P_1 nonconforming element and Morley element*, Computational Sciences for 21-century (John Bristeau et al, ed.), Wiley & Sons, 1997, pp. 189–198.
7. Zhenghui Xie, *Domain decomposition and multigrid methods for nonconforming plate elements*, Ph.d dissertation, The Institute of Computation Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1996.
8. X. Zhang, *Two-level additive Schwarz methods for the biharmonic problem discretized by conforming C^1 elements*, SIAM J. Numer. Anal. **34** (1997), 881–904.

INSTITUTE OF COMPUTATIONAL MATHEMATICS, CHINESE ACADEMY OF SCIENCES, P.O.Box 2719, BEIJING 100080, CHINA

E-mail address: shi@lsec.cc.ac.cn

LASG, INSTITUTE OF ATMOSPHERIC PHYSICS, CHINESE ACADEMY OF SCIENCES, P.O.Box 2718, BEIJING 100080, CHINA

E-mail address: xzh@lasgsgia4.iap.ac.cn

Additive Schwarz Methods for Hyperbolic Equations

Yunhai Wu, Xiao-Chuan Cai, and David E. Keyes

1. Introduction

In recent years, there has been gratifying progress in the development of domain decomposition algorithms for symmetric and nonsymmetric elliptic problems and even some indefinite problems. Many methods possess the attractive property that the convergence rate is optimal, i.e., independent of the size of the discrete problem and of the number of subdomains, or within a polylog factor of optimal. There is, in comparison, relatively little in the domain decomposition literature on hyperbolic problems. Quarteroni [8, 9] used nonoverlapping domain decomposition methods based on the spectral collocation approximation on systems of conservation laws. Gastaldi and Gastaldi [5, 6] set up a nonoverlapping domain decomposition scheme based on the finite element approximation for the transport equation. These contributions establish the boundary operators that lead to well-posed decoupled problems, which can then be discretized and solved by standard means.

Our interests in this paper are rather different. We examine overlapping domain decomposition preconditioners, and leave the original global discretization fully in tact. Rather than deriving interface conditions that lead to decomposed solutions that are mathematically equivalent (to within some specified discretization tolerance) to the solutions of the undecomposed problem, we derive an approximate inverse that can be applied in a concurrent manner, subdomain-by-subdomain, and that effectively preconditions the original undecomposed operator, whose action is already trivial to apply in the same concurrent manner. There seem to have been to date no such additive or multiplicative Schwarz preconditioners leading to optimal convergence rates for hyperbolic equations.

Based on the standard Galerkin method [4] an ASM algorithm is formulated. The preconditioned problems are solved by the GMRES method. The convergence

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 65F10, 65N30, 65Y05.

Supported in part by the Lingnan Foundation, by the National Natural Science Foundation, and by NASA contract NAGI-1692 while the author was in residence at Old Dominion University.

Supported in part by NSF grant ECS-9527169, and by NASA contract NAS1-19480 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2911.

Supported in part by NSF grant ECS-9527169 and by NASA Contracts NAS1-19480 and NAS1-97046 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2911.

rate is shown to be asymptotically independent of the time and space mesh parameters and the number of subdomains, provided that the time step is fine enough, namely of such a size as would be typical for temporal stability reasons in an explicit discretization. As these limits are exceeded, numerical experiments based on a Galerkin discretization show a rapid deterioration in convergence rate. (Upwinded discretizations permit explicit stability limits to be exceeded, in the sense that the resulting preconditioned iterations on each time step can converge sufficiently rapidly to be cost-effective in comparison with explicit methods, as discussed in a forthcoming sequel.) Convergence rate is experimentally observed to be relatively independent of overlap.

Just as in the parabolic case, but in contrast to the elliptic case, no coarse-level mesh is required in forming an optimal preconditioner. Good speedups are available on a distributed-memory machine, as would be expected of a problem with a purely local preconditioner.

2. Model problem

We consider for convenience the constant-coefficient linear scalar hyperbolic equation:

$$(1) \quad \frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} = 0, \quad \text{in } \Omega \times I,$$

together with proper boundary and initial conditions, where Ω is a bounded domain in R^2 with boundary Γ and $I = (0, T)$ is a time interval.

All results in this paper extend without difficulty to the more general linear hyperbolic problem:

$$\frac{\partial u}{\partial t} + \operatorname{div}(bu) + cu = f, \quad \text{in } (x, t) \in \Omega \times I,$$

where Ω is a bounded domain in R^d ($d = 2$ or 3), the coefficients $b = (b_1, \dots, b_d)$ and c depend smoothly on (x, t) , and $\frac{1}{2}\operatorname{div}b + c \geq c_0 \geq 0$ in $\Omega \times I$, for stability.

By implicit temporal finite differencing, we obtain the following problem:

$$(2) \quad \left\{ \begin{array}{ll} -\tau_k \left(\frac{\partial u_k}{\partial x} + \frac{\partial u_k}{\partial y} \right) + u_k = f, & \text{in } \Omega \\ u_k = g, & \text{on } \Gamma_- \end{array} \right\}, k = 1, 2, \dots, K,$$

where τ_k is the k^{th} time step, K is the number of steps, $\sum_{k=1}^K \tau_k = T$, $f = u_{k-1}$, and Γ_- is the inflow boundary defined by

$$\Gamma_- = \{(x, y) \in \Gamma : n(x, y) \cdot \beta < 0\},$$

where $n(x, y)$ is the outward unit normal to Γ at the point $(x, y) \in \Gamma$, and $\beta = (-\tau_k, -\tau_k)$. Any implicit multistep time-integration method leads to a system like (2), in which f more generally contains a linear combination of the solution at earlier time steps.

The following notation will be used throughout this chapter:

$$\begin{aligned} & \langle u, v \rangle_- = \int_{\Gamma_-} uv(n \cdot \beta) ds, \quad \langle u, v \rangle_+ = \int_{\Gamma_+} uv(n \cdot \beta) ds, \\ & \langle u, v \rangle = \int_{\Gamma} uv(n \cdot \beta) ds, \quad |u|_{\beta} = (\int_{\Gamma} u^2 |n \cdot \beta| ds)^{\frac{1}{2}}, \\ & \|u\| = \|u\|_{L_2(\Omega)}, \quad |u| = (\int_{\Gamma} u^2 ds)^{\frac{1}{2}}, \end{aligned}$$

where $\Gamma_+ = \Gamma \setminus \Gamma_- = \{(x, y) \in \Gamma : n(x, y) \cdot \beta \geq 0\}$.

3. Standard Galerkin method

Let us consider the standard Galerkin method for the problem (2), which can be given the following variational formulation: Find $u \in H^1(\Omega)$, such that

$$(3) \quad (u_\beta + u, v)_- < u, v >_- = (f, v)_- < g, v >_-, \quad \forall v \in H^1(\Omega),$$

where we omit the subscript k , and where $u_\beta = -\tau(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y})$. By Green's formula, it is easy to show that

$$(u_\beta, v) = < u, v > - (u, v_\beta).$$

The stability of (3) is a consequence of the following property of the bilinear form $B_\beta(u, v) \equiv (u_\beta + u, v)_- < u, v >_-$:

$$B_\beta(u, u) = \|u\|^2 + \frac{1}{2}|u|_\beta^2.$$

The symmetric part of $B_\beta(u, v)$ is

$$A_\beta(u, v) = (u, v) + \frac{1}{2}(< u, v >_+ - < u, v >_-),$$

and the skew-symmetric part is

$$S_\beta(u, v) = \frac{1}{2} < u, v > - (u, v_\beta).$$

Define the β -norm as $\|\cdot\|_\beta = \sqrt{B_\beta(\cdot, \cdot)}$.

We choose $V^h \subset H^1(\Omega)$ as a finite element space of continuous piecewise polynomial functions of degree one or higher on a mesh of quasi-uniform element size h . We discretize equation (3) in space by the Galerkin finite element method and have the approximation: Find $u^h \in V^h$ at each time step k , such that

$$(4) \quad B_\beta(u^h, v^h) = (f, v^h)_- < g, v^h >_-, \quad \forall v^h \in V^h.$$

We require the following assumption for the theoretical analysis:

ASSUMPTION 1. The relation between τ and h is

$$\tau \leq Ch^{1+s},$$

where $s \geq 0$.

In the case of velocity magnitudes different from unity in (1), Assumption 1 becomes a CFL condition, and the allowable time step must be reduced in inverse proportion to the global maximum of the velocity.

We have some lemmas pertaining to B_β , A_β , and S_β as follows.

LEMMA 2. *There exist positive constants c_1 and c_2 , independent of τ , such that*

$$\begin{aligned} |B_\beta(u, v)| &\leq c_1 \|u\|_\beta \cdot \|v\|_\beta, \quad \forall u, v \in V^h(\Omega), \\ B_\beta(u, u) &\geq c_2 \|u\|_\beta^2, \quad \forall u \in V^h(\Omega). \end{aligned}$$

LEMMA 3. *There exist positive constants c_3 and c_4 , independent of τ , such that*

$$\begin{aligned} |A_\beta(u, v)| &\leq c_3 \|u\|_\beta \cdot \|v\|_\beta, \quad \forall u, v \in V^h(\Omega), \\ A_\beta(u, u) &\geq c_4 \|u\|_\beta^2, \quad \forall u \in V^h(\Omega). \end{aligned}$$

LEMMA 4. *There exists a constant $c_5 > 0$, independent of τ , such that*

$$|S_\beta(u, v)| \leq c_5 \tau \left(\frac{1}{h} \|u\| \|v\| + |u| |v| \right) \leq c_5 (h^s \|u\| \|v\| + |u| |v|), \quad \forall u, v \in V^h(\Omega).$$

An additive Schwarz algorithm for (4) is formulated following [2]. Let $\Omega_i, i = 1, \dots, N$, be nonoverlapping subregions of Ω with quasi-uniform diameters H , such that $\bigcup \bar{\Omega}_i = \bar{\Omega}$. The vertices of any Ω_i not on $\partial\Omega$ coincide with the fine-grid mesh vertices. We define an overlapping decomposition of Ω , denoted by $\{\Omega'_i, i = 1, \dots, N\}$, by extending each Ω_i to a larger region Ω'_i , which is cut off at the physical boundary of Ω . The overlap is generous in the sense that there exists a constant $\alpha > 0$ such that $\text{dist}(\partial\Omega'_i \cap \Omega, \partial\Omega'_i \cap \Omega) \geq \alpha H, \forall i$.

Corresponding to the domain decomposition, we decompose the finite element space V^h at each time step k in the customary manner [2], i.e., $V^h = V_1^h + \dots + V_N^h$, where V_k^h is a discrete space whose support is confined to the extended subdomain Ω'_i .

The basic building blocks of the algorithm, projection operators $Q_i : V^h \rightarrow V_i^h, i = 1, \dots, N$, are defined by

$$(5) \quad B_\beta(Q_i u^h, v^h) = B_\beta(u^h, v^h), \quad \forall v^h \in V_i^h.$$

The subproblems have homogeneous Dirichlet boundary conditions for the interior boundary. We can introduce the operator $T = Q_1 + \dots + Q_N$ and form the transformed linear system

$$(6) \quad Tu^h = b,$$

where the right-hand side is defined by $b \equiv Tu^h = \sum_{i=1}^N Q_i u^h$, which can be computed without the knowledge of u^h by solving the subproblems (5).

If T is invertible, we show below that equation (6) has the same, unique solution as (4). The operator T is inconvenient to obtain explicitly, but the action of T on a function in V^h is straightforward to compute, consisting of independent problems in subdomains. Thus the preconditioned form (6) can be solved by a Krylov iterative method, such as GMRES [10].

With Assumption 1 and the inverse inequalities

$$(7) \quad \|u\|_1 \leq \frac{c}{h} \|u\|,$$

and (from [7])

$$(8) \quad |u| \leq C \sqrt{\|u\| \cdot \|u\|_1},$$

we have

$$\begin{aligned} \|u\|_\beta^2 &\leq \|u\|^2 + \frac{1}{2}\tau C|u|^2 \\ &\leq \|u\|^2 + C\tau\|u\| \cdot \|u\|_1 \\ &\leq \|u\|^2 + Ch^s\|u\|^2 \\ &\leq C\|u\|^2. \end{aligned}$$

(assuming that $h \leq 1$). On the other hand, we obtain,

$$(9) \quad \|u\|^2 \leq \|u\|_\beta^2,$$

which leads to:

LEMMA 5. *The β -norm is equivalent to the L_2 norm.*

Therefore, following [2], we come to the conclusion that:

LEMMA 6. *There exists a constant $C_0 > 0$, independent of h and H such that, for all $u^h \in V^h$, there exist $u_i^h \in V_i^h$ with $u^h = \sum_{i=1}^N u_i^h$, and $\sum_{i=1}^N \|u_i^h\|_\beta^2 \leq C_0^2 \|u^h\|_\beta^2$. C_0 generally depends upon the subdomain overlap α .*

We give an estimate in the following lemma for the skew-symmetric part $S_\beta(\cdot, \cdot)$, which shows that the skew-symmetric part is a lower order term compared with the symmetric part, and can therefore be controlled.

LEMMA 7. *There exists a constant δ , $0 < \delta < 1$, independent of τ , h , and H , such that*

$$|S_\beta(u^h, Tu^h)| \leq \delta B_\beta(u^h, Tu^h), \forall u^h \in V^h.$$

PROOF. We use the inequalities (7) and (8) throughout the proof.

By the definition of Q_i , $i = 1, \dots, N$, we have

$$B_\beta(Q_i u^h, Q_i u^h) = B_\beta(u^h, Q_i u^h),$$

and furthermore

$$\|Q_i u^h\|_{\beta(\Omega'_i)} \leq C \|u^h\|_{\beta(\Omega'_i)}.$$

Following Lemma 4, Lemma 5, and Assumption 1, we can show

$$|S_\beta(Q_i u^h, u^h - Q_i u^h)| \leq Ch^s \|u^h\|_{\beta(\Omega'_i)}^2.$$

Using Lemma 2, Lemma 5, and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \|u^h\|_\beta^2 &\leq \sum_{i=1}^N B_\beta(Q_i u^h, u_i^h) \\ &\leq c \sum_{i=1}^N \|Q_i u^h\|_\beta \cdot \|u_i^h\|_\beta \\ &\leq c \sqrt{\sum_{i=1}^N \|Q_i u^h\|_\beta^2} \cdot \sqrt{\sum_{i=1}^N \|u_i^h\|_\beta^2} \\ &\leq c C_0 \sqrt{B_\beta(u^h, Tu^h)} \cdot \|u^h\|_\beta, \end{aligned}$$

and hence we obtain

$$\|u^h\|_\beta^2 \leq CB_\beta(u^h, Tu^h),$$

which finally leads to the conclusion. \square

We can summarize the following main result:

THEOREM 8. (a) *There exist constants $c > 0$ and $C > 0$, independent of τ , h and H , such that*

$$C\|u^h\|_\beta \geq \|Tu^h\|_\beta \geq c\|u^h\|_\beta, \quad \forall u^h \in V^h.$$

(b) *There exists a constant $C(\delta) > 0$, such that $\forall u^h \in V^h$*

$$A_\beta(u^h, Tu^h) \geq C(\delta)\|u^h\|_\beta^2.$$

Since the symmetric part of the preconditioned linear system is positive definite, GMRES will converge at a rate that is asymptotically independent of h , H , and τ .

TABLE 1. Convergence rate dependence on time-step exponent s

s	It.	Time
0.5	4.0	16.71s
0.1	7.4	26.25s
0.0	10.0	33.95s
-0.1	44.1	146.95s

4. Numerical Results

The preceding theorems are useful in motivating effective algorithms but leave unanswered quantitative questions about the magnitudes of constants in part (a) of Theorem 8 about the extent of dependence of $C(\delta)$ on the size of the overlap in parts (b) of the same theorems, and about the sensitivity of results to inexact solutions in the subdomains. The latter is important since inexactness is usually a practical requirement. For these reasons, we include some numerical experiments, whose purpose is to quantify the dependence of the convergence rate on potentially “bad” parameters, including time step exponent, subdomain overlap, inexactness, overall problem size, and number of subdomains into which the problem is decomposed.

We first vary s between the very conservative $s = \frac{1}{2}$, down to the Courant limit of $s = 0$, and a little beyond into negative values. We solve model problem (1) with backward Euler time-stepping on a uniform grid with central-differencing. We hold the problem size fixed at $h^{-1} = 512$, implying approximately one-quarter of a million degrees of freedom overall, and the the number of subdomains at $p = 16$, arranged in a 4×4 decomposition, with 128×128 grid cells owned by each subdomain. The overlap between subdomains is one mesh cell. We demand a reduction of 10^{-5} in relative residual norm at each time step, accomplished by linear subiterations of GMRES with a subdomain preconditioner of ILU(0).

In Table 1, we tabulate the number of linear iterations per time step, averaged over 10 consecutive steps, and also the execution time for these ten time steps, as measured on the Intel Paragon, with one processor per subdomain. It is evident that the theoretical restriction on the time step to the Courant limit is necessary for reasonable conditioning of the linear iterations.

In Table 2 we vary the subdomain overlap in the preceding example, using two different subdomain preconditioners, exact solvers (indicated by “LU”), and inexact solvers of zero-fill incomplete LU-type (“ILU”). For ILU, three different values of s are tried, hovering around the Courant limit. Convergence criteria and iteration counts are as before. The overlap is tabulated in terms of the thickness of the overlap region in number of cells all around each subdomain, except where cut off at the boundary. We observe that increasing overlap has a slightly beneficial effect when it alone is the bottleneck to better convergence, as in the LU situation. In the practical ILU case, overlap beyond a minimum of one has little to no effect on the convergence rate, provided reasonable values of s are employed. In the case of negative s , increasing the overlap actually causes the convergence rate to deteriorate.

Comparing the first and third result columns, we see that inexactness has a price of approximately a factor of two in convergence rate. In practice, this does not

TABLE 2. Convergence rate dependence on subdomain overlap

overlap	LU, $s = 0$	ILU, $s = 0.1$	ILU, $s = 0$	ILU, $s = -0.1$
1	5.0	7.4	10.0	44.1
2	3.0	7.7	10.0	45.2
4	3.0	7.4	10.0	48.5

TABLE 3. Convergence rate dependence on number of subdomains, and fixed-size parallel scalability

p	per node	It.	Time	Time/It.	Rel. Sp.	Rel. Sp./It.
4	256×256	10.0	171.54s	1.715s		
16	128×128	10.0	33.95s	0.339s	5.05	5.05
64	64×64	10.2	10.41s	0.102s	16.48	16.81

translate into any advantage for exact solvers since the convergence criterion at each time step would usually be commensurate with the temporal truncation error, and looser than that employed here, and the cost for computing an exact factorization of a coefficient matrix on each time step cannot be amortized in practical time-dependent problems (though it could be in (1)).

For Table 3, we fix $s = 0$, the overlap at 1, and the subdomain preconditioner as ILU(0). We perform a problem-size-fixed scaling analysis at $h^{-1} = 512$ by employing successively more subdomains, in going from 4 to 16 to 64 processors. Note that the problem size on each processor decreases by a factor of 2 in each of the x and y directions in this scaling. As before we tabulate the average number of iterations per time step averaged over 10 steps, and the execution time for first ten time steps. The execution time is also presented per iteration, and the speedups (relative to four processors) are presented for both overall time and for time per iteration. This allows for separate measurement of “numerical scalability” of the algorithm and “implementation scalability” of the software/hardware system, with any deterioration of convergence rate at highly granular decompositions factored out.

Our main observations are the virtual independence of convergence rate on the number of subdomains p , for s at the Courant limit, as predicted by the theory, and the better than linear parallel scalability. The latter phenomenon is due to the increasingly good reuse of data in the working set required by the subdomain solvers as the problem-per-processor shrinks. This is a well-known effect in memory-limited machines. Because of the insensitivity of the convergence rate to decomposition, the two speedup measurements are nearly identical.

Table 4 is similar to Table 3; in fact, the last line of each tabulates the same execution, and both run over the same number of processors, except that Table 4 runs a problem small enough to fit on one processor, which grows in size as the number of processors grows. This is known as a Gustafson scaling analysis. It is a practical scaling for large-scale applications and it has the advantage of keeping the workingset per node constant over a range of problem size and processor number.

TABLE 4. Convergence rate dependence on number of subdomains, and Gustafson parallel scalability

p	h^{-1}	It.	Time	Time/It.	Rel. Eff.	Rel. Eff./It.
1	64	8.0	6.60s	0.083s		
4	128	10.1	8.94s	0.089s	0.74	0.93
16	256	10.2	10.00s	0.098s	0.66	0.84
64	512	10.2	10.41s	0.102s	0.63	0.81

The one-subdomain case is special (and would have converged in one iteration had we employed an LU solver). In tabulating efficiency, we take the ratio of the execution times on the successively scaled problems. The efficiency can be viewed as the incremental efficiency of the last processor added, when loaded with the same work per processor. Presenting the relative efficiency per iteration is more important in this case, since the iteration count does degrade in going from one to many subdomains.

Our main observation is that the efficiency remains very high, almost explicit-like. There is no coarse grid to bottleneck this method. On the other hand the frequent global inner products are minor bottlenecks.

We employed the Portable Extensible Toolkit for Scientific Computing (PETSc) [1] from Argonne National Laboratory for the numerical studies.

5. Conclusions

We have used the standard Galerkin method and to formulate an optimal additive Schwarz method for general scalar linear hyperbolic equations. The same techniques leading to optimal convergence rates for the parabolic and elliptic cases have been used here, after identification of the proper norm. The method of proof does not permit evaluation of the key constants in the theory.

The theoretical techniques employed here may be applicable to other equations, e.g., linearized Euler equations and hyperbolic systems of conservation laws, after transformation to canonical form and operator splitting. We are currently pursuing such extensions.

Because of Assumption 1 limiting the size of τ , the implicit method described herein might not appear to offer any advantage relative to the correspondingly spatially discretized temporally explicit method, which has equally good or better parallelization properties, and would not require iteration on each time step to solve a linear system. On the other hand, temporal truncation accuracy limits the algebraic accuracy required in the solution of the implicit system to just a few matrix-vector products, and the implicit form may be thought of as a defect-correction solver. Two practical applications of the results of this paper may be to: (1) problems with multiple scales, with some scales finer than the explicit stability limit, all of which could be treated implicitly with this method, and (2) problems with embedded hyperbolic regions, for which a uniform Schwarz preconditioned framework is desired. We mention [3] as an example.

References

1. S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *PETSc Users' Manual*, Tech. Rpt. ANL-95/11, Mathematics and Computer Science Division, Argonne National Laboratory, 1995.
2. X.-C. Cai, *Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations*, Ph.D thesis, Tech. Rep. 461, Courant Institute, 1989.
3. X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, *Parallel Newton-Krylov-Schwarz Algorithms for the Transonic Full Potential Equation*, SIAM J. Sci. Comput. **19** (1998), 246–265.
4. P. Ciarlet, The finite element method for elliptic problems, North-Holland, Amsterdam, 1978.
5. L. Gastaldi, *A domain decomposition for the transport equation*, Proc. of the Sixth Int. Conf. on Domain Decomposition Methods (A. Quarteroni, et al., eds.), AMS, Providence, 1992, pp. 97–102.
6. ——, *On a domain decomposition for the transport equation: Theory and finite element approximation*, IMA J. Num. Anal. **14** (1994), 111–136.
7. P. Lesaint, *Finite element methods for symmetric hyperbolic equations*, Numer. Math. **21** (1973), 244–255.
8. A. Quarteroni, *Domain decomposition methods for systems of conservation laws: Spectral collocation approximations*, SIAM J. Sci. Stat. Comput. **11** (1990), 1029–1052.
9. ——, *Domain decomposition methods for wave propagation problems*, in Domain-based Parallelism and Problem Decomposition Methods in Computational Science and Engineering (D. E. Keyes, Y. Saad, and D. G. Truhlar, eds.), SIAM, Philadelphia, 1995, pp. 21–38.
10. Y. Saad and M. H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7** (1986), 856–869.

DEPARTMENT OF COMPUTER SCIENCE, ZHONGSHAN UNIVERSITY, GUANG ZHOU 510275, P. R. CHINA

E-mail address: wu@cs.odu.edu

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF COLORADO AT BOULDER, BOULDER, CO 80309-0430

E-mail address: cai@cs.colorado.edu

COMPUTER SCIENCE DEPARTMENT, OLD DOMINION UNIVERSITY, NORFOLK, VA 23529-0162 & ICASE, NASA Langley Res. Ctr., HAMPTON, VA 23681-2911

E-mail address: keyes@icase.edu

Part 4

Applications

A Minimum Overlap Restricted Additive Schwarz Preconditioner and Applications in 3D Flow Simulations

Xiao-Chuan Cai, Charbel Farhat, and Marcus Sarkis

1. Introduction

Numerical simulations of unsteady three-dimensional compressible flow problems require the solution of large, sparse, nonlinear systems of equations arising from the discretization of Euler or Navier-Stokes equations on unstructured, possibly dynamic, meshes. In this paper we study a highly parallel, scalable, and robust nonlinear iterative method based on the Defect Correction method (DeC), the Krylov subspace method (Krylov), the minimum overlap restricted additive Schwarz method (RAS), and the incomplete LU factorization technique (ILU). To demonstrate the robustness of the method, we test the capability of the DeC-Krylov-RAS solver for several flow regimes including transonic and supersonic flows around an oscillating wing and a moving aircraft. The parallel scalability is also tested on a multiprocessor computer. We consider the unsteady 3D Euler's equation

$$(1) \quad \frac{\partial W}{\partial t} + \operatorname{div}(F(W)) = 0$$

with certain initial and boundary conditions. Unstructured mesh and variable time stepping are used in our numerical simulation, however, for the sake of simplicity in the discussion, we let Δt and h be the fixed time and spatial discretization parameters, and $\Phi_h^{(2nd)}$ a second order MUSCL discretization of $\operatorname{div}(F(\cdot))$. A fully discretized scheme, which is of second order in both space and time, can be written as

$$(2) \quad \frac{3W_h^{n+1} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1}) = 0.$$

Here n is a running time step index and W_h^0 is the given initial solution. Assuming that W_h^n and W_h^{n-1} are known, (2) is a large, sparse, nonlinear algebraic system of equations that has to be solved at every time step to a certain accuracy.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 76D05, 65Y05.

Key words and phrases. Restricted additive Schwarz, 3D compressible flows, Unstructured meshes, Parallel performance.

This research was supported in part by NSF ECS-9725504 and AFOSR F49620-97-1-0059.

2. Discretization and the nonlinear solver

We are interested in applying the method of DeC-Krylov-RAS to the system of Euler's equation:

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x} F_1(W) + \frac{\partial}{\partial y} F_2(W) + \frac{\partial}{\partial z} F_3(W) = 0,$$

where $W = (\rho, \rho u, \rho v, \rho w, E)^T$ and $(F_1, F_2, F_3)^T$ is the convective flux as defined in [8]. Here and in the rest of the paper ρ is the density, $U = (u, v, w)^T$ is the velocity vector, E is the total energy per unit volume, and p is the pressure. These variables are related by the state equation for a perfect gas

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho \|U\|^2 \right),$$

where γ denotes the ratio of specific heats ($\gamma = 1.4$ for air).

The computational domain is discretized by a tetrahedral grid. We use unstructured grids since they provide flexibility for tessellating complex, moving geometries and for adapting to flow features, such as shocks and boundary layers. We locate the variables at the vertices of the grid, which gives rise to a cell-vertex scheme. The space of solutions is taken to be the space of piecewise linear continuous functions. The discrete system is obtained via a finite volume formulation; see e.g., Koobus and Farhat [10]. We determine the n th time step size Δt^n in the following way. Let CFL be a pre-selected positive number. For each vertex x_i , let h_i be the size of the control volume centered at x_i , and we define the local time step size by

$$\Delta t_i^n = h_i \frac{\text{CFL}}{C_i + \|U_i\|_2}$$

and then the global time step is defined by

$$(3) \quad \Delta t^n = \min_i \{\Delta t_i^n\}.$$

Here C_i is the sound speed, and U_i is the velocity vector.

One of the effective techniques for solving (2) is based on the so-called Defect Correction (DeC) method ([11]): Suppose that we have an initial guess $W_h^{n+1,0}$ for W_h^{n+1} obtained by using information calculated at previous time steps, we iterate for $j = 0, 1, \dots$,

$$(4) \quad W_h^{n+1,j+1} = W_h^{n+1,j} + \xi^j,$$

where ξ^j is the solution of the following linear system of equations

$$(5) \quad \left(\frac{3}{2\Delta t} I + \partial_W \Phi_h^{(1st)}(W_h^n) \right) \xi^j = - \left(\frac{3W_h^{n+1,j} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1,j}) \right).$$

Here I is an identity matrix and $\Phi_h^{(1st)}(\cdot)$ is a first order MUSCL discretization of $\text{div}(F(\cdot))$. To simplify the notation, we use

$$g_{n+1,j} \equiv - \left(\frac{3W_h^{n+1,j} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1,j}) \right)$$

to denote the *nonlinear residual* at the j th DeC iteration of the $(n+1)$ th time step and re-write (5) as

$$(6) \quad A_n \xi^j = g_{n+1,j}.$$

We remark that (2) doesn't have to be solved exactly. All we need is to drive the nonlinear residual to below a certain *nonlinear tolerance* $\tau > 0$, i.e.,

$$(7) \quad \|g_{n+1,j}\|_2 \leq \tau \|g_n\|_2$$

such that $W_h^{n+1,j}$ gives a second order accurate solution in both space and time. Also (6) does not need to be solved very accurately either, as its solution provides only a search direction for the outer DeC iteration. Preconditioned iterative methods are often used for finding a $\xi^j = M_n^{-1} \eta^j$ such that

$$(8) \quad \|A_n M_n^{-1} \eta^j - g_{n+1,j}\|_2 \leq \delta \|g_{n+1,j}\|_2$$

for certain *linear tolerance* $\delta > 0$. Here M_n^{-1} is a preconditioner for A_n .

The effectiveness of the above mentioned method depends heavily, among other things, on the choice of the preconditioner and a balanced selection of the nonlinear and linear stopping tolerance τ and δ . In this paper, we focus on the study of a parallel restricted additive Schwarz preconditioned iterative method for solving (6) with various δ . More discussions and computational experience with the selection of the nonlinear and linear stopping tolerance τ and δ can be found in [2].

3. RAS with minimum overlap

We now describe a version of the RAS preconditioner, which was recently introduced in [3], with the smallest possible non-zero overlap. We consider a sparse linear system

$$(9) \quad A\xi = g,$$

where A is an $n \times n$ nonsingular sparse matrix obtained by discretizing a system of partial differential equations, such as (1), on a tetrahedral mesh $\mathcal{M} = \{K_i, i = 1, \dots, M\}$, where K_i are the tetrahedra. Using an element-based partitioning, \mathcal{M} can be decomposed into N nonoverlapping sets of elements, or equivalently into N overlapping sets of nodes (since tetrahedra in different subsets may share the same nodes). Let us denote the node sets as $W_i, i = 1, \dots, N$. Let W be the set of all the nodes, then we say that the node-based partition

$$W = \bigcup_{i=1}^N W_i$$

is a minimum overlap partition of W . “minimum” refers to the fact that the corresponding element-based partition has zero overlap. The nodes belonging to more than one subdomains are called interface nodes. To obtain a node-based nonoverlapping partition, we identify a unique subdomain as the sole owner of each interface node. This leads to a node-based nonoverlapping partition of W , as shown in Fig.1 for a 2D mesh, or more precisely $W_i^{(0)} \subset W_i$, and

$$\bigcup_{i=1}^N W_i^{(0)} = W \text{ and } W_i^{(0)} \cap W_j^{(0)} = \emptyset \text{ for } i \neq j.$$

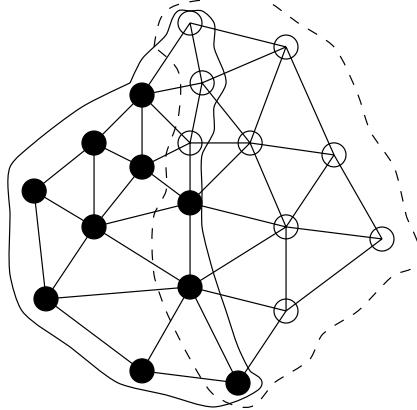


FIGURE 1. A minimum overlap two-subdomain partition. $W_1^{(0)}$ contains all the ‘●’ nodes, and $W_2^{(0)}$ contains all the ‘○’ nodes, therefore $W_1^{(0)} \cap W_2^{(0)} = \emptyset$. $W_1^{(1)}$ contains all the nodes bounded inside the solid curve, and $W_2^{(1)}$ contains all the nodes bounded inside the dotted curve.

Let m be the total number of nodes in W . Associated with each W_i^0 we define a restriction operator R_i^0 . In matrix terms, R_i^0 is an $m \times m$ block-sub-identity matrix whose diagonal blocks are set to $I_{5 \times 5}$ if the corresponding node belongs to W_i^0 and to a zero 5×5 block otherwise. Similarly we can define R_i for each W_i . Note that both R_i^0 and R_i are of size $n \times n$. With this we define the matrix,

$$A_i = R_i A R_i .$$

Note that although A_i is not invertible, we can invert its restriction to the subspace

$$A_i^{-1} \equiv ((A_i)_{|L_i})^{-1} ,$$

where L_i is the vector space spanned by the set W_i in \mathbf{R}^n . Recall that the regular additive Schwarz (AS) preconditioner is defined as $M_{AS}^{-1} = \sum R_i A_i^{-1} R_i$, e.g., [4, 13]. Our RAS algorithm can be simply described as follows: Obtain the solution $\xi = M_{RAS}^{-1} \eta$ by solving the right-preconditioned system

$$A M_{RAS}^{-1} \eta = g$$

with a Krylov subspace method, where the preconditioner is defined by

$$M_{RAS}^{-1} \equiv R_1 A_1^{-1} R_1^0 + \cdots + R_N A_N^{-1} R_N^0 .$$

In the numerical experiments to be reported in the next section, all subdomain problems are solved with ILU(0) and GMRES(5) ([12]) is used as the Krylov solver. Because of the page limit, we shall restrict our discussion to this particular preconditioner. Other issues can be found in the papers [1, 2]. We remark that the action of R_i^0 to a vector does not involve any communication in a parallel implementation, but R_i does. As a result, RAS is cheaper than AS in terms of the communication cost. We will show in the next section that RAS is in fact also cheaper than AS in terms of iteration counts.

TABLE 1. Iteration counts. Euler flow passing an oscillating wing at Mach 0.89. The mesh contains $n = 22014$ nodes. $subd$ is the number of subdomains and δ is the stopping condition.

$n = 22014$	$\delta = 10^{-2}$			$\delta = 10^{-8}$			
	$subd$	JAC	AS	RAS	JAC	AS	RAS
4	26	7	6	129	40	30	
8	26	8	6	129	41	30	
16	26	9	7	129	44	31	
32	26	9	6	129	46	32	

4. Numerical studies

In this section, we present several numerical simulations of unsteady 3D flows to demonstrate the scalability and robustness of the RAS preconditioner. We also include some comparisons with the regular additive Schwarz method and the simple pointwise Jacobi method (JAC). Note that a point in the mesh represents an 5×5 block matrix. Other recent development in the application of RAS in CFD can be found in [6, 9].

4.1. Parallel implementation issues. We implemented the algorithm on a number of parallel machines, and the top-level message-passing calls are implemented through MPI [7]. We partition the mesh by using the TOP/DOMDEC package [5]. We require that all subdomains have more or less the same number of mesh points. An effort is made to reduce the number of mesh points along the interfaces of subdomains to reduce the communication cost. The mesh generation and partitioning are considered as pre-processing steps, and therefore not counted toward the CPU time reported. The sparse matrix defined by (5) is constructed at every time step and stored in an edge-based sparse format.

4.2. A transonic flow passing a flexible wing. We tested our algorithm for an Euler flow passing a flexible wing at $M_\infty = 0.89$. The wing is clamped at one end and forced into the harmonic motion. We test the algorithm on two unstructured meshes with 22014 and 331233 nodes, respectively. The two meshes are generated independently, i.e., one is not a refined version of the other.

We focus on the performance of the algorithm for solving a single linear system. The results on the coarser grid are summarized in Table 1 with CFL=900. Table 2 is for the finer mesh with CFL=100. Due to the special choice of the CFL numbers, the time steps for the two test cases are roughly the same. Comparing the RAS columns in Tables 1 and 2, we see that there is little dependence on the mesh sizes. And, we also see clearly that JAC has a strong dependence on the mesh sizes. As the number of subdomains grows from 4 to 16 or 32, the number of iterations of RAS stays more or less the same without having a coarse space in the preconditioner. Another observation is that RAS requires 20% to 30% fewer number of iterations than AS for the test cases.

4.3. A supersonic flow passing a complete aircraft. We consider a supersonic, $M_\infty = 1.9$, Euler flow passing a complete aircraft. The mesh contains $n = 89144$ nodes. In Table 3, we report the number of iterations for solving a single

TABLE 2. Iteration counts. Euler flow passing an oscillating wing at Mach 0.89. The mesh contains $n = 331233$ nodes. $subd$ is the number of subdomains and δ is the stopping condition.

$n = 331233$	$\delta = 10^{-2}$			$\delta = 10^{-8}$			
	$subd$	JAC	AS	RAS	JAC	AS	RAS
4	58	9	7	253	51	36	
8	58	9	7	253	52	36	
16	58	10	7	253	52	36	

TABLE 3. Supersonic Euler's flow on a 3D unstructured mesh at Mach 1.90. The number of processors equals the number of subdomains $subd$. Number of nodes = 89144. The CPU (in seconds) time below is for solving one linear system.

$n = 89144$	$subd = 4$			$subd = 8$		
	JAC	AS	RAS	JAC	AS	RAS
ITER	96	37	28	96	38	29
CPU	33	29	23	16	14	11
COMM	0.2	0.15	0.1	0.3	0.2	0.15

linear system with JAC, AS and RAS preconditioned GMRES(5) methods. The CPU and communication (COMM) times are obtained on a SGI Origin 2000 with 4 and 8 processors. Even though this is a shared memory machine, we still treat it as a message-passing machine. The results are given in Table 3 for $\delta = 10^{-6}$ and CFL=1000.

5. Concluding remarks

We studied the performance of a newly introduced RAS preconditioner and tested it in several calculations including a transonic flow over an oscillating wing and a supersonic flow passing a complete aircraft. RAS compares very well against the regular additive Schwarz method in terms of iteration counts, CPU time and communication time when implemented on a parallel computer. Even though we do not have a coarse space, the number of iterations is nearly independent of the number of subdomains for all the test cases.

References

1. X.-C. Cai, M. Dryja, and M. Sarkis, *A convergence theory for restricted additive Schwarz methods*, in preparation, 1998.
2. X.-C. Cai, C. Farhat, B. Koobus, and M. Sarkis, *Parallel restricted additive Schwarz based iterative methods for general aerodynamic simulations*, in preparation, 1998.
3. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, Tech. Report CU-CS-843-97, Department of Computer Science, University of Colorado at Boulder, 1997.
4. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), 604–620.

5. C. Farhat, S. Lanteri, and H. Simon, *TOP/DOMDEC: A software tool for mesh partitioning and parallel processing and applications to CSM and CFD computations*, Comput. Sys. Engrg. **6** (1995), 13–26.
6. W. Gropp, D. Keyes, L. McInnes, and M. Tidriri, *Parallel implicit PDE computations: Algorithms and software*, Proceedings of Parallel CFD'97, A. Ecer, et al., edt., Manchester, UK, 1997., 1997, to appear.
7. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI – Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, 1995.
8. C. Hirsch, *Numerical Computation of Internal and External Flows, Vol I*, Wiley, New York, 1990.
9. D. Keyes, D. Kaushik, and B. Smith, *Prospects for CFD on petaflops systems*, CFD Review 1997, M. Hafez et al., edt., 1997, to appear.
10. B. Koobus and C. Farhat, *Time-accurate schemes for computing two- and three-dimensional viscous fluxes on unstructured dynamic meshes*, AIAA-96-2384, 1996.
11. R. Martin and H. Guillard, *A second order defect correction scheme for unsteady problems*, Computers and Fluids **25** (1996), 9–27.
12. Y. Saad and M. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7** (1986), 856–869.
13. B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF COLORADO AT BOULDER, BOULDER, CO 80309.

E-mail address: cai@cs.colorado.edu

DEPARTMENT OF AEROSPACE ENGINEERING, UNIVERSITY OF COLORADO AT BOULDER, BOULDER, CO 80309.

E-mail address: charbel@alexandra.colorado.edu

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF COLORADO AT BOULDER, BOULDER, CO 80309.

E-mail address: msarkis@cs.colorado.edu

Time Domain Decomposition for European Options in Financial Modelling

Diane Crann, Alan J. Davies, Choi-Hong Lai, and Swee H. Leong

1. Introduction

Finance is one of the fastest growing areas in modern applied mathematics with real world applications. The interest of this branch of applied mathematics is best described by an example involving shares. Shareholders of a company receive dividends which come from the profit made by the company. The proceeds of the company, once it is taken over or wound up, will also be distributed to shareholders. Therefore shares have a value that reflects the views of investors about the likely dividend payments and capital growth of the company. Obviously such value will be quantified by the share price on stock exchanges. Therefore financial modelling serves to understand the correlations between asset and movements of buy/sell in order to reduce risk. Such activities depend on financial analysis tools being available to the trader with which he can make rapid and systematic evaluation of buy/sell contracts. There are other financial activities and it is not an intention of this paper to discuss all of these activities. The main concern of this paper is to propose a parallel algorithm for the numerical solution of an European option.

This paper is organised as follows. First, a brief introduction is given of a simple mathematical model for European options and possible numerical schemes of solving such mathematical model. Second, Laplace transform is applied to the mathematical model which leads to a set of parametric equations where solutions of different parameteric equations may be found concurrently. Numerical inverse Laplace transform is done by means of an inversion algorithm developed by Stehfest [4]. The scalability of the algorithm in a distributed environment is demonstrated. Third, a performance analysis of the present algorithm is compared with a spatial domain decomposition developed particularly for time-dependent heat equation. Finally, a number of issues are discussed and future work suggested.

2. European Options

One simple and interesting financial model known as the European Option has two types of contracts available namely, call options and put options. The holder

1991 *Mathematics Subject Classification*. Primary 65D99; Secondary 65M06.

Key words and phrases. Time Domain Decomposition, Financial Applications.

Scalability tests in this paper were performed by the first two authors.

of a call option has the right, at a prescribed time known as the expiry date, to purchase a prescribed asset for a prescribed amount usually known as the strike price. While the other party of the contract must sell the asset if the holder chooses to buy it. On the other hand, the holder of a put option has the right, at the expiry date, to sell the prescribed asset at the strike price. While the other party of the contract must buy the asset if the holder chooses to sell it [6]. This section only examines a European call option. The stochastic background of the equation is not discussed in this paper and readers should consult [6].

Let $v(x, t)$ denotes the value of an option where x is the current value of the underlying asset and t is the time. The value of the option depends on $\sigma(t)$, E , T and $r(t)$ which are, respectively, known as the volatility of the underlying asset, the strike price, the expiry time and the interest rate. The Black-Scholes analysis for one independent variable leads to the famous Black-Scholes equation [6],

$$(1) \quad \frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} + rx \frac{\partial v}{\partial x} - rv = 0 \in \Omega^+$$

where $\Omega^+ = \{x : x \geq 0\}$. In order to describe a European call option, boundary conditions and final conditons are required. Since the call option is worthless at $x = 0$ even if there is a long time to reach expiry, therefore it is sensible to have $v(0, t) = 0$. Since the asset price increases without bound, therefore it becomes likely that the option will be exercised and the magnitude of the strike price becomes less important. Therefore it is sensible to have $v(s, t) \sim s$ as $s \rightarrow \infty$. At expiry, if $x > E$ then one should exercise the call option, i.e. to hand over an amount E to obtain an seest with x . However, if $x < E$ at expiry, one should not exercise the call option. Since the expiry date is in the future, the final condition $v(x, T) = \max(x - E, 0)$ must be imposed. The solution v for $t < T$ is required.

The financial interpretation of the above model is as follows. First, the difference between the return on an option portfolio, which involves the first two terms, and the return on a bank deposit, which involves the last two terms, should be zero for a European option. Second, the only parameter that affects the option in a stochastic way is the volatility $\sigma(t)$ which measures the standard deviation of the returns.

Since (1) is a backward equation, one can transform it to a forward equation by using $\tau = T - t$ and it leads to,

$$(2) \quad \frac{\partial v}{\partial \tau} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 v}{\partial x^2} + rx \frac{\partial v}{\partial x} - rv \in \Omega^+$$

subject to boundary conditions $v(0, \tau) = 0$ and $v(s, \tau) \sim s$ as $s \rightarrow \infty$ and initial conditions $v(x, 0) = \max(x - E, 0)$.

An analytic solution may be derived if a change of variable is made where the Black-Scholes equation is converted to a time-dependent heat conduction equation with constant coefficients [6]. However a field method, such as finite volume methods, is of more interest for two reasons. First, there are many examples in multi-factor models such that a reduction of the time dependent coefficient to a constant coefficient heat equation is impossible. Hence analytic form of solutions cannot be found. Second, the computational environment at Greenwich is based on the finite volume code PHYSICA [1] which is the main research and development code for multi-physics work. The code has capability of solving unsteady diffusion, convection and radiation type of equations. Financial modelling typically requires large number of simulations and hence computing resources and efficiency of

algorithms are very important in order to make evaluation and decision before the agreement of a contact. With the present day high performance computing and/or distributed computing, parallel algorithms offer efficient numerical solutions to the equation given by (2).

3. Time Domain Decomposition

For time varying $\sigma(t)$ and $r(t)$, it is possible to make suitable coordinate transformation to the Black-Scholes equation in order to obtain a time independent like heat equation [6]. Hence the method described in this section may then be applied. Here, a method is described which focuses on time independent coefficients σ and r . Taking Laplace transform of (2) and taking integration by parts to the left-hand-side of the transformed equation, one obtains the parametric equation

$$(3) \quad \frac{1}{2}\sigma^2x^2\frac{d^2u}{dx^2} + rx\frac{du}{dx} - (r + \lambda_j)u = v(x, 0) \in \Omega^+$$

subject to boundary conditions $u(0; \lambda_j) = 0$ and $u(s; \lambda_j) = \frac{s}{\lambda_j}$. Here $u(x, \lambda_j)$ is the Laplace transform of $v(x, t)$ and λ_j is a discrete set of transformation parameters defined by

$$(4) \quad \lambda_j = j\frac{\ln 2}{\tau}, \quad j = 1, 2, \dots, m$$

where m is required to be chosen as an even number [5]. An approximate inverse Laplace transform [4] may be used to retrieve $v(x, t)$ according to

$$(5) \quad v(x, \tau) \approx \frac{\ln 2}{\tau} \sum_{j=1}^m w_j u(x, \lambda_j)$$

where

$$w_j = (-1)^{m/2+j} \sum_{k=(1+j)/2}^{\min(j, m/2)} \frac{k^{m/2}(2k)!}{(m/2-k)!k!(k-1)!(j-k)!(2k-j)!}$$

is known as the weight factor. Each of the above m parametric equations may be rewritten as

$$(6) \quad \frac{d}{dx}\left(\frac{1}{2}\sigma^2x^2\frac{du}{dx} + (r - \sigma^2)xu\right) - (2r - \sigma^2 + \lambda_j)u = v(x, 0)$$

The computational domain has a uniform mesh and finite volume method is applied to (6) which leads to

$$(7) \quad \int_S \left(\frac{1}{2}\sigma^2x^2\frac{du}{dx} + (r - \sigma^2)xu \right) ds - \int_\Omega (2r - \sigma^2 + \lambda_j)ud\Omega = \int_\Omega v(x, 0)d\Omega$$

The resulting system of linear equations is solved in a local area network which consists of P workstations. There are two possible implementations as follow.

First, the solution at a particular time τ is being sought. For the case when $m = P$, one would expect ideal load balancing. The total computing time, t_{A_1} , using the present scheme can be estimated as $t_A = t_1 + t_a/P$ where t_1 is the computing time for solving one parametric equation and t_a is the corresponding computing time for numerical inverse Laplace transforms given by (5) and is assumed to be equally spreaded across the P workstations. For the case when $m > P$, one would expect just a slight out of load balance for the reason that m is

possibly not an integral multiple of P . It is possible to estimate the total computing time as

$$(8) \quad t_{A_1} = \lceil \frac{m}{P} \rceil t_1 + t_a / P$$

The case $m < P$ is not of interest because the active workstations become a subset of the local area network.

Second, the solutions at P particular times τ_k , $k = 1, 2, \dots, P$ are being sought. In this situation, each workstation looks after the solutions of m parametric equations and the corresponding inverse Laplace transform at a particular time τ_k . Hence the total computing time, t_{A_2} , may be estimated as

$$(9) \quad t_{A_2} = mt_1 + t_a$$

In order to check the scalability of the algorithm, a cell-centred finite volume scheme is applied to the constant coefficient heat equation,

$$(10) \quad \nabla^2 u = \frac{1}{k} \frac{\partial u}{\partial t} \in -1 < x < 1, -1 < y < 1$$

subject to unit boundary conditions along the whole boundary and zero initial condition. A uniform 16 x 16 grid where the set of discrete equation is solved by Gauss-Seidel iteration. Solutions at eight time values, $\tau = 0.1, 0.2, 0.5, 1, 2, 4, 10$ and 20, are sought. For the purpose of demonstration, a network of 4 T800 transputers were used as the hardware platform. The computing times for $P = 1, 2$ and 4 are respectively 2537, 1309 and 634 seconds. The speed-up ratio for using two and four processors are thus 1.94 and 4 respectively. More results about this test problem can be found in [2] and experience shown in the paper suggests that $m = 8$ provides sufficient accuracy for the model test. Note that the value of m determines the accuracy of the inverse Laplace transform and hence it depends on the mesh size or the number of grid point. In general, $m = \sqrt{N}/2$ where N is the total number of grid points in a two-dimensional problem [2]. Note also that the scalability property as shown in this section also applies to eqn (6).

4. A Comparison with Spatial Domain Decomposition

In order to find out the suitability of the proposed algorithm for a distributed computing environment, a comparison with a spatial domain decomposition method is examined in this section. The spatial domain decomposition method is similar to the one developed by Dawson et al [3] for unsteady heat conduction equation. The problem described in (2) is partitioned into P subdomains so that a coarse mesh of mesh size $H = s/P$ is imposed with interior boundary of the subdomains being the same as the nodal points of the coarse mesh. In order to determine the interior boundary values of each of the subdomains, an explicit scheme derived from using a central difference method along the spatial axis and a forward difference method along the temporal axis is as follows,

$$(11) \quad v_i^n = (\frac{1}{2} \sigma^2 x_i^2 \frac{\Delta t}{H^2} - rx_i \frac{\Delta t}{2H}) v_{i-1}^{n-1} + (\frac{1}{2} \sigma^2 x_i^2 \frac{\Delta t}{H^2} + rx_i \frac{\Delta t}{2H}) v_{i+1}^{n-1} \\ + (1 - \sigma^2 x_i^2 \frac{\Delta t}{H^2} - r \Delta t) v_i^{n-1}$$

Here subscripts i denote the mesh points on the coarse mesh and superscripts n denotes the time step at $t = n\Delta t$. The choice of Δt must satisfy the coarse grid restriction for an explicit scheme which is

$$(12) \quad \Delta t \leq \min_{x_i} \left(\frac{\sigma^2 x_i^2}{H^2} + r \right)^{-1} \leq (\sigma^2(P-1)^2 + r)^{-1}$$

Note here that Δt should be of the order of h^2 where h is the grid size of the fine mesh. Therefore the total number of time steps involved in the present calculation is $T(\sigma^2(P-1)^2 + r)$.

It is reasonable to assume that the computing time for obtaining the solution at a new time step using the fine mesh and a finite difference scheme is the same as the computing time for obtaining the solution of a parametric equation given in (3). Therefore the computing time for marching one time step forward using the classical spatial domain decomposition with P subdomains on P processors is t_1/P . The total parallel computing time can be estimated as

$$(13) \quad t_B = T(\sigma^2(P-1)^2 + r) \left(\frac{t_1}{P} + \frac{t_b}{P} \right)$$

where t_b is the overheads for obtaining interior boundary conditons and is assumed to be equally spreaded across the P workstations.

It is natural to require $t_{A_1} < t_B$ for any advantage of the proposed time domain decomposition scheme to be happened when comparing with the classical spatial domain decomposition, and hence one would require $\lceil \frac{m}{P} \rceil t_1 < \frac{T(\sigma^2(P-1)^2 + r)}{P} t_1$, i.e.

$$(14) \quad \lceil \frac{m}{P} \rceil < \frac{T(\sigma^2(P-1)^2 + r)}{P}$$

When m is an integral multiple of P , one obtains

$$(15) \quad m < T(\sigma^2(P-1)^2 + r)$$

In other words, one requires the number of parameters in inverse Laplace transforms to be smaller than the number of time steps involved in spatial domain decomposition. Note that the dominant term in the inequality is obviously T . Since typical values of m is usually much smaller than s/h for an acceptable accuracy of inverse Laplace transform [2], the inequality (15) is easily satisfied with typical ranges of $0.05 < \sigma < 0.45$ and $0.6 < r < 1.1$. Therefore for a given value of P , the inequality offers only a very mild restriction. Hence the time domain decomposition method proposed in this paper has advantage over the classical spatial domain decomposition method for European call options.

From (15), we have $mt_1 < T(\sigma^2(P-1)^2)\frac{t_1}{P}$ which combines with (13) to give

$$(16) \quad \frac{t_{o_1}}{t_o} < \frac{T(\sigma^2(P-1)^2 + r)}{P}$$

The ratio governs the number of time levels τ_k to be allowed in each workstation in order that the inequality $t_{A_2} < t_B$ remains valid. Supposing each parametric equation has N grid points involved in the discretisation, the total number of floating point operations involved in (5) can be easily counted as $2N + Nm$. Also, supposing some of floating point operations in (11) can be done once for all, the total number of floating point operations for the update of interior boundary conditions

is counted as $18P$. Hence (16) become

$$(17) \quad m < \frac{18T}{N}(\sigma^2(P-1)^2 + r) - 2$$

It can be checked that the inequality is easily satisfied with the above typical ranges of σ and r .

5. Conclusions

A parallel algorithm based on Laplace transform of the time domain into a set of parametric equations is developed for European call option. Distributed computing may be applied to solve the parametric equations concurrently. An inverse Laplace transform based on Stehfest method is applied to retrieve the solution. The method is compared with classical spatial domain decomposition. A preliminary analysis shows that the proposed method has advantage over the spatial domain decomposition method.

References

1. Centre for Numerical Modelling and Process Analysis - University of Greenwich, *Physica - user guide*, 1995, Beta Release 1.0.
2. AJ Davies, J Mushtaq, Radford LE, and Crann D, *The numerical Laplace transform solution method on a distributed memory architecture*, Applications of High Performance Computing in Engineering V (H Power and JJ Casares Long, eds.), Computational Mechanics Publications, 1997, pp. 245–254.
3. CN Dawson, Q Du, and TF Du Pont, *A finite difference domain decomposition algorithm for the numerical solution of the heat equation*, Math Comp **57** (1991), 63–67.
4. H Stehfest, *Numerical inversion of Laplace transforms*, COMM ACM **13** (1970), 47–49.
5. D V Widder, *The Laplace transform*, Princeton University Press, Princeton, 1946.
6. P Wilmott, S Howison, and J Dewynne, *The mathematics of financial derivatives*, Press Syndicate of the University of Cambridge, New York, 1995.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HERTFORDSHIRE, HATFIELD, UK
E-mail address: D.Crann@herts.ac.uk

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HERTFORDSHIRE, HATFIELD, UK
E-mail address: A.J.Davies@herts.ac.uk

SCHOOL OF COMPUTING & MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH,
 WELLINGTON STREET, WOOLWICH, LONDON SE18 6PF, UK
E-mail address: C.H.Lai@greenwich.ac.uk

QUANTITATIVE RESEARCH AND TRADING GROUP, THE CHASE MANHATTAN BANK, 125 LONDON
 WALL, LONDON EC2Y 5AJ, UK
E-mail address: Swee.Leong1@chase.com

Parallel Modal Synthesis Methods in Structural Dynamics

Jean-Michel Cros

1. Introduction

Within the finite element framework, we deal with large-scale eigenvalue problems induced by free vibration analysis of complex structures. The classical approach for the solution of such problems consists first of reducing the number of unknowns, allowing to reduce the computational cost of eigensolver, because only the lowest eigenfrequencies are classically researched in modal analysis. Component mode synthesis (CMS) or dynamic substructuring methods are appropriate tools for this reduction. In this paper we will discuss about the parallel implementation of CMS methods. We consider, in particular, among several CMS methods [4], the fixed-interface method which we briefly recall. Assuming that the studied domain is partitioned in N_s non-overlapping substructures, the global solution of the eigenvalue problem to be solved in the domain can be written as the sum of the local solutions (*fixed interface modes*) to elasticity eigenproblem to be solved in each subdomain clamped on the interface, and extensions (*constraint modes or coupling modes*), in each subdomain, of functions which represent the motion of the interface. The dynamic behavior of the global structure can be approximated in the low frequencies range by truncating the series which represent the different spaces. It remains to define what kind of Dirichlet's conditions has to be prescribed for the coupling of the substructures. The most classical choice [3] consists of prescribing at the discrete level the shape functions spanning the interfacial interpolation space. In this case, all the motions of the interface are represented, but, the number of *constraint modes* is equal to the number of degrees of freedom (d.o.f.) of the interface. From the parallel point of view the constraint and the fixed interface modes can be computed independently in each subdomain. An other choice, proposed by Bourquin [1], consists of filtering information in order to represent the interface's motion only in the low frequency range, thanks to a spectral decomposition of the interface operator coupling subdomains. However, the computation of the *coupling modes* involves all the subdomains. We propose a parallel implementation of these methods [5] thanks to the use of techniques arising in domain decomposition.

1991 *Mathematics Subject Classification.* Primary 73K12; Secondary 49R10, 65N25.

Key words and phrases. Eigenvalues, Modal Synthesis Methods, Domain Decomposition, Iterative Solver, Parallel Computing.

Acknowledgements - We would like to acknowledge O.N.E.R.A. for computing facilities.

The paper is organized as follows : Section 2 presents different parallel implementations of the Craig and Bampton (CB) method, one of them enabling to avoid the costly computation of the constraint modes. Section 3 describes some fixed interface methods using coupling modes. In Section 4, some numerical results obtained on the Intel PARAGON machine are presented. We also compare, from the cpu time point of view, the results obtained thanks to a parallel sparse eigensolver.

2. Parallel implementation of the Craig and Bampton method

The model reduction of each substructure (s) is given by the projection of local mass ($M^{(s)}$) and rigidity ($K^{(s)}$) matrices onto the Ritz basis :

$$\begin{Bmatrix} u_i^{(s)} \\ u_b^{(s)} \end{Bmatrix} = [\Phi^{(s)} \quad \Psi^{(s)}] \begin{Bmatrix} \eta^{(s)} \\ u_b^{(s)} \end{Bmatrix} \text{ with } \Phi^{(s)} = \begin{bmatrix} \phi^{(s)} \\ 0 \end{bmatrix}, \Psi^{(s)} = \begin{bmatrix} -K_{ii}^{(s)-1} K_{ib}^{(s)} \\ I \end{bmatrix}$$

where the subscripts i and b respectively refer to the internal and boundary d.o.f., $\eta^{(s)}$ is the vector of normal modes intensities, $\Psi^{(s)}$ and $\Phi^{(s)}$ are respectively the constraint and fixed interface modes. By assembling the model reduction of each substructure we get the following reduced eigenproblem :

$$(1) \quad \tilde{K}z = \lambda \tilde{M}z$$

with the reduced rigidity (\tilde{K}) and mass matrices (\tilde{M}) given by :

$$\tilde{K} = \begin{bmatrix} \Omega_\alpha^{(s)^2} & \\ 0 & S \end{bmatrix}, \tilde{M} = \begin{bmatrix} I & \Phi^{(s)^T} M^{(s)} \Psi^{(s)} \\ \Psi^{(s)^T} M^{(s)} \Phi^{(s)} & \sum_{s=1}^{N_s} \Psi^{(s)^T} M^{(s)} \Psi^{(s)} \end{bmatrix}$$

The reduced stiffness matrix includes Schur complement matrix (S) and $\Omega_\alpha^{(s)^2}$ which is a diagonal matrix storing the squares of the subdomains eigenvalues. The reduced mass matrix includes static condensed mass matrices and the inertia coupling between the constraint and fixed interface modes. The size of (1) is equal to the sum of the number of fixed interface modes, chosen in each subdomain, and the interface's number of d.o.f. (which in the case of complex structures is large). This "reduced" eigenproblem is still too large to take benefit from eigensolver such QR's method and claims the use of subspace algorithm (Lanczos, ...). The generalized eigenproblem is then reduced to a standard form and involves matrix-vector products with $\tilde{K}^{-1} \tilde{M}$. Different implementations have been studied in order to reduce the computational cost thanks to the use of parallel machine with distributed memory. The first one (*solver I*), is the most natural, because it uses the fact that fixed interface modes and constraint modes can be computed independently in each subdomain. Then, a processor is in charge of a subdomain and the local reduced matrices are built in parallel. However the reduced eigenproblem (1) is assembled and solved in sequential way. The second one (*solver II*), looks like Hybrid Craig and Bampton method, proposed by Farhat and Gérardin [7], but consists in a primal formulation. The Schur complement matrix is not assembled and the action of S^{-1} on a vector is done by using iterative Schur complement method [8] with different acceleration techniques [5] (generalized Neumann-Neumann preconditioner, technique for taking into account multiple right hand sides, ...). The main task, consists in the computation of the constraint modes in order to build \tilde{M} . This operation may be avoided. For this purpose a third method (*solver III*) allows for the computation of matrix-vector products with reduced mass matrix in an implicit way. The details of this operation are as

follows. The product $\tilde{M}^{(s)}z^{(s)}$ (with any vector z) can be written in matrix form as :

$$(2) \quad \begin{aligned} & \left[\begin{array}{cc} I & \Phi^{(s)T} M^{(s)} \Psi^{(s)} \\ \Psi^{(s)T} M^{(s)} \Phi^{(s)} & \Psi^{(s)T} M^{(s)} \Psi^{(s)} \end{array} \right] \begin{Bmatrix} z_1^{(s)} \\ z_2^{(s)} \end{Bmatrix} \\ & = \begin{Bmatrix} z_1^{(s)} + \Phi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)} \\ \Psi^{(s)T} M^{(s)} \Phi^{(s)} z_1^{(s)} + \Psi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)} \end{Bmatrix} \end{aligned}$$

The description of the algebraic operations requires the knowledge of the different components of this vector, without any explicit assembling of the matrices $[\Phi^{(s)T} M^{(s)} \Psi^{(s)}]$, $[\Psi^{(s)T} M^{(s)} \Psi^{(s)}]$, $[\Psi^{(s)T} M^{(s)} \Phi^{(s)}]$. First of all, for each subdomain, the following matrix-vector product is computed :

$$(3) \quad \begin{bmatrix} M_{ii}^{(s)} & M_{ib}^{(s)} \\ M_{bi}^{(s)} & M_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \phi^{(s)} \\ 0 \end{bmatrix} = \begin{bmatrix} M_{ii}^{(s)} \phi^{(s)} \\ M_{bi}^{(s)} \phi^{(s)} \end{bmatrix}$$

Then, at each iteration of a given eigensolver, the operations described below are done :

Computation of $\Psi^{(s)T} M^{(s)} \Phi^{(s)} z_1^{(s)}$

1. Compute $M_{ii}^{(s)} \phi^{(s)} z_1^{(s)}$ with (3).
2. Solve Dirichlet's problem in $\Omega^{(s)}$ with zero on $\Gamma^{(s)}$ (interface with other subdomains) and external force :

$$(4) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ 0 \end{Bmatrix}$$

3. Compute the forces induced by the opposite displacement, solution of (4) :

$$(5) \quad \begin{bmatrix} K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} -K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} -K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ 0 \end{Bmatrix}$$

4. Compute $M_{bi}^{(s)} \phi^{(s)} z_1^{(s)}$ with (3).
5. Assemble interface's contributions of vectors obtained at the two previous steps :

$$(6) \quad \left\{ M_{bi}^{(s)} \phi^{(s)} z_1^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \right\}$$

Computation of $\Psi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)}$

1. Solve Dirichlet's problem in $\Omega^{(s)}$ with $z_2^{(s)}$ on $\Gamma^{(s)}$:

$$(7) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ z_2^{(s)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

2. Compute the matrix-vector product, the local mass matrix $M^{(s)}$, solution of problem (7) :

$$(8) \quad \begin{bmatrix} M_{ii}^{(s)} & M_{ib}^{(s)} \\ M_{bi}^{(s)} & M_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} -K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} \\ z_2^{(s)} \end{Bmatrix} = \begin{Bmatrix} (-M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{ib}^{(s)}) z_2^{(s)} \\ (-M_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{bb}^{(s)}) z_2^{(s)} \end{Bmatrix}$$

3. Solve Dirichlet's problem in $\Omega^{(s)}$ with zero on $\Gamma^{(s)}$ and external forces :

$$(9) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} (-M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{ib}^{(s)}) z_2^{(s)} \\ 0 \end{Bmatrix}$$

4. Compute the forces induced by opposite displacement, solution of (9) :

$$(10) \quad \begin{aligned} & \left[\begin{array}{cc} K_{bi}^{(s)} & K_{bb}^{(s)} \end{array} \right] \left\{ \begin{array}{c} (K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \\ 0 \end{array} \right\} \\ & = \left\{ (K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \right\} \end{aligned}$$

5. Assemble interface contributions of vectors (10) and (8). We get then :

$$\left\{ (M_{bb}^{(s)} - M_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \right\}$$

Computation of $\Phi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)}$

1. Computation of the forces arising from the solution of (7) :

$$(11) \quad \begin{aligned} & \left[\begin{array}{cc} \phi^{(s)T} M_{ii}^{(s)} & \phi^{(s)T} M_{ib}^{(s)} \end{array} \right] \left\{ \begin{array}{c} -K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} \\ z_2^{(s)} \end{array} \right\} \\ & = \left\{ -\phi^{(s)T} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} + \phi^{(s)T} M_{ib}^{(s)} z_2^{(s)} \right\} \end{aligned}$$

The reduced matrix-vector product is achieved through the assembling of the contributions of each subdomain. Let us note that all the matrices (except the mass matrix) are already used by the iterative Schur complement method.

3. Fixed interface methods with coupling modes

With regard to the previous method, a small set of modes is required for the coupling of the substructures. The new Ritz basis [1] is indeed defined by :

$$\left\{ \begin{array}{l} u_i^{(s)} \\ u_b^{(s)} \end{array} \right\} = \left[\begin{array}{cc} \Phi^{(s)} & \Psi^{(s)} \end{array} \right] \left\{ \begin{array}{l} \eta^{(s)} \\ \xi \end{array} \right\} \text{ with } \psi^{(s)} = R^{(s)} \psi \text{ and } \Psi^{(s)} = \left[\begin{array}{c} -K_{ii}^{(s)-1} K_{ib}^{(s)} \psi^{(s)} \\ \psi^{(s)} \end{array} \right]$$

where ψ are the first normal modes of the Schur complement matrix (interface modes), $R^{(s)}$ is the restriction matrix of the global interface (Γ) to the local interface ($\Gamma^{(s)}$), ξ is the vector of the coupling modes intensities and $\Psi^{(s)}$ defines the coupling modes. Thanks to this definition of coupling modes, \tilde{K} is diagonal, and the size of the reduced eigenproblem is equal to the sum of the number of fixed interface modes chosen in each subdomain and the number (N_Γ) of coupling modes. Then, the solution of the reduced eigenproblem is easy. The difficulty is now to find the first normal modes of the interface's operator, in the present case the Schur complement matrix (*method 1*) with or without mass condensed matrix (denoted by B).

$$(12) \quad S u_\Gamma = \lambda_\Gamma B u_\Gamma \quad \rightarrow \quad S^{-1} B u_\Gamma = \frac{1}{\lambda_\Gamma} u_\Gamma$$

From the computation view point, the method presents an interest, when the interface is large, only if the Schur complement is not assembled. Once again, the iterative Schur complement method (S is *not explicitly* inverted) is used. Various acceleration techniques have been used to reduce as best as possible the cost of this operation (Section 2). We have considered different mass matrices : identity, lumped ($B = \sum_{s=1}^{N_s} M_{bb}^{(s)}$) and static condensed mass matrix. For the latter, the matrix-vector product can be done without any assembling operation (Section 2). Nevertheless, the computation of the first normal modes of the Schur complement is

a costly operation. The use of an approximated inverse [2] (*method 2*) of the Schur complement matrix (such that $T \approx S^{-1}$) avoids to consider an iterative method, because only matrix-vector product with T are then required.

$$(13) \quad TBu_\Gamma = \frac{1}{\lambda_\Gamma} u_\Gamma$$

A good inverse of the Schur complement matrix is clearly the Neumann-Neumann preconditioner, written here in its basic form :

$$(14) \quad T = \sum_{s=1}^{N_s} R^{(s)^T} S^{(s)^{-1}} R^{(s)}$$

For sake of generality, it would be required to take into account decompositions with cross-points and floating subdomains. In the case of cross-points, [2] defines a new extension operator (reflection and cut-off function) which possess an interpretation at the continuous level. We propose to improve (14) by averaging the contributions of each subdomain through the introduction of weighting matrices [8]. On the other hand, the global problem can be well posed (no rigid body modes), but the decomposition may lead to local Neumann's problems not well posed (no Dirichlet's conditions). Shifting [2] the global problem enables to erase the floating subdomains, however one can also handle directly with them thanks to a filter operator which stems from balancing method [9]. Finally, the approximated inverse is given by :

$$(15) \quad T = (I - G(G^T SG)^{-1}G^T S) \sum_{s=1}^{N_s} R^{(s)^T} P^{(s)} S^{(s)^+} P^{(s)} R^{(s)}$$

where $P^{(s)}$, stands for the weighting matrix, defined such that $\sum P^{(s)} = I/\Gamma$, G is the rectangular matrix which stores the interface restriction of the local rigid body motions, and $S^{(s)^+}$ is the inverse of the projection of the image of $S^{(s)}$ in $\mathbb{R}^{N_{lfron}^{(s)}}$, where $N_{lfron}^{(s)}$ denotes the local interface size. Let us note, that the rigid body motions have to be also considered as coupling modes because, by construction, they are perpendicular to u_Γ .

4. Numerical results

In order to present some numerical experiments, we consider a three-dimensional beam. The finite element discretization consists in 3000 hexahedral Q1-Lagrange elements (11253 d.o.f.); the mesh is cut in 8 boxes ($2 \times 2 \times 2$); the size of the interface is thus large (2253 d.o.f.) and each substructure possess 1728 d.o.f. Ten eigenpairs are required. The computation is carried out on 8 nodes of the Intel Paragon.

Table 1 reports the cpu times of the main tasks of the CB's method. As we can see, with solver III, the cpu time compared with that of solver I is reduced of 65%. Concerning the method 1, we plot (Fig.1) the relative error committed on the global eigenfrequencies versus the number (N_Γ) of coupling modes. We have compared the results for different mass matrices, used during the computation of the coupling modes. For static condensed mass matrix the improvement is obvious : with 10 coupling modes, the relative error is less than 1%. With lumped and identity mass the graphs are nearly the same, but more coupling modes are included in the approximation's space to get the same relative error. To understand these results,

TABLE 1. 3d beam (2-2-2) - cpu times of the parallel implementations of the Craig and Bampton method

	constraint modes		
	with solver I	solver II	without solver III
constraint modes	80,2s	80,2s	Ø
fixed interface modes	90,4s	90,4s	90,4s
assembling reduced eigenproblem	267,7s	133,9s	Ø
solving reduced eigenproblem	70,1s	71,3s	86,8s
restitution	0,6s	0,6s	1,6s
total	508,8s	376,2s	177,2s

TABLE 2. 3d beam (2-2-2) - method 1 - coupling modes computing

Mass	Identity		Lumped		Static condensed
N_Γ	20	30	20	30	20
Lanczos	117,2s	167,2s	126,9s	176,7s	138,2s
Matrix (mass)-vector	0,007s	0,01s	7,9s	10,9s	24,7s

TABLE 3. 3d beam (2-2-2) - cpu times

solver III	method 1 ($N_\Gamma = 15$)	method 2 ($N_\Gamma = 25$)	parallel sparse eigensolver
177s	190s	130s	97s

we plot (Fig. 2) the spectrum of the Schur complement matrix with different mass matrices along the interface. As we can see, the spectrum of the Schur complement, when static condensed mass is used, is very closed to this of the global structure. Then, knowing the excitation's range of the global structure it seems possible to define a by-passing frequency criterion, allowing to select enough coupling modes during their computations to get a good approximation of the global eigenvalues.

Table 2 reports the cpu times corresponding to the main operation (method 1) : the computation of the coupling modes. Taking into account the static condensed mass is not too expensive. Despite the use of acceleration techniques (Fig. 3) (such as coarse grid induced by rigid modes, preconditioner, techniques enabling to take into account multiple right hand sides), the computation of the coupling modes remains a cpu times consuming operation.

Now we compare (Table 3) the cpu times required to find 10 eigenpairs with different techniques. From the cpu times point of view, despite the different implementations proposed, in particular for the CB's method, parallel sparse eigensolver [6], based on Lanczos algorithm and nonoverlapping domain decomposition method, is the fastest method. This result was predictable because in the three methods (solver III, method 1 and parallel sparse eigensolver) the same interface problem (Fig. 3) is solved, and CMS needs additional operations (such that fixed interface modes computation, restitution of the solution in each subdomain). In addition, parallel sparse eigensolver provides the best approximation of the eigenpairs. By another way, for all methods, parallel scalability [6] is guaranteed thanks to the iterative substructuring method, if load balancing is correct.

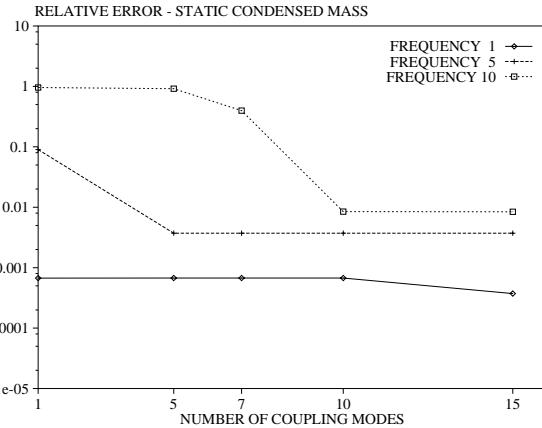
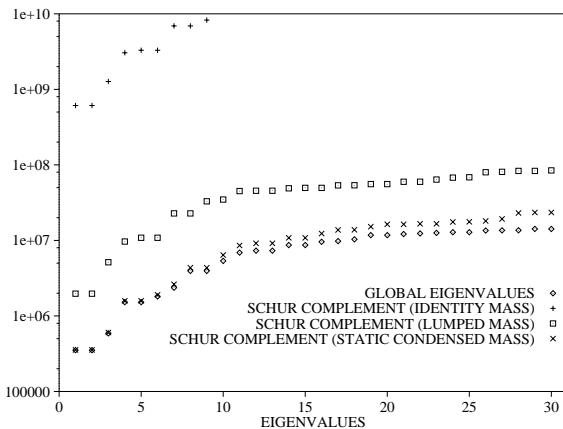
FIGURE 1. Error for beam 3d, method 1, N_F increasing

FIGURE 2. Beam 3d, eigenspectrum of the Schur complement with different mass matrices

By the way, method 2 is shown to be less accurate (with 25 coupling modes the relative error is less than 3%) than method 1 (see also [2, 10]), but quickly gives an approximation of the global eigenpairs with regard to the others CMS.

In conclusion, the methods presented in this paper are particularly well suited to the architecture of parallel computers. The different methods can be improved in order to take into account other finite elements (plates, shells). A comparison with other methods using coupling modes [2, 10] seems necessary. Let us lastly note that particular methods can be derived for special cases (decomposition without cross-point).

References

1. F. Bourquin and F. d'Hennezel, *Application of domain decomposition techniques to modal synthesis for eigenvalue problems*, Proc. Fifth Int. Conf. on Domain Decomposition Meths. (Philadelphia) (D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds.), 1992.

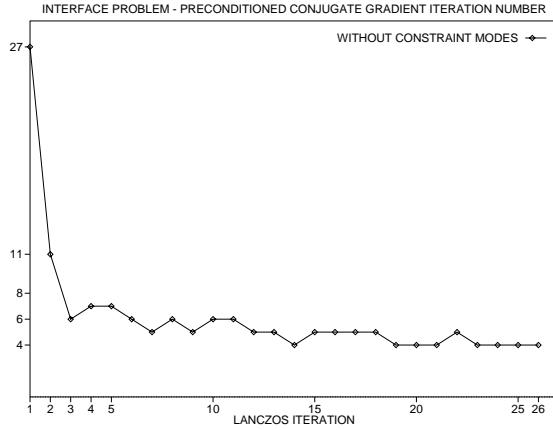


FIGURE 3. Beam 3d, solver III, solution of the reduced eigenvalue problem

2. F. Bourquin and R. Namar, *Decoupling and modal synthesis of vibrating continuous systems*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (P. Bjørstad, M. Espedal, and D. Keyes, eds.), Wiley and Sons, 1996.
3. R.R. Craig and M.C.C. Bampton, *Coupling of substructures for dynamic analysis*, AIAA Journal **6** (1968), 1313–1319.
4. R.R. Craig, Jr, *A review of time domain decomposition and frequency domain component mode synthesis methods*, Combined Experimental/Analytical Modeling of Dynamic Structural systems, vol. 67, ASME, 1985.
5. J.M. Cros, *Résolution de problèmes aux valeurs propres en calcul des structures par utilisation du calcul parallèle*, Ph.D. thesis, Ecole normale supérieure de Cachan, 1997.
6. J.M. Cros and F. Léne, *Parallel iterative methods to solve large-scale eigenvalue problems in structural dynamics*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (Bergen), 1996.
7. C. Farhat and M. Gérardin, *On a component mode synthesis method and its application to incompatible substructures*, Computer and Structures **51** (1994), 459–473.
8. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances, no. 2, North-Holland, 1994.
9. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg. **9** (1993), 233–241.
10. D. Rixen, *Substructuring and dual methods in structural analysis*, Ph.D. thesis, Université de Liège, 1997.

LABORATOIRE DE MODÉLISATION ET MÉCANIQUE DES STRUCTURES, UNIVERSITÉ PIERRE ET MARIE CURIE, 4 PLACE JUSSIEU, F75252 PARIS CEDEX 05
E-mail address: cros@ccr.jussieu.fr

Efficient Computation of Aerodynamic Noise

Georgi S. Djambazov, Choi-Hong Lai, and Koulis A. Pericleous

1. Introduction

Computational Fluid Dynamics codes based on the Reynolds averaged Navier-Stokes equations may be used to simulate the generation of sound waves along with the other features of the flow of air. For adequate acoustic modeling the information about the sound sources within the flow is passed to a linearized Euler solver that accurately resolves the propagation of sound through the non-uniformly moving medium.

Aerodynamic sound is generated by the flow of air or results from the interaction of sound with airflow. Computation of aerodynamic noise implies *direct* simulation of the sound field based on first principles [6]. It allows complex sound fields to be simulated such as those arising in turbulent flows.

When building a software tool for this simulation two options exist: (a) to develop a new code especially for this purpose, or (b) to use an existing Computational Fluid Dynamics (CFD) code as much as possible. (As it will be shown later, due to numerical diffusion conventional CFD codes tend to smear the sound signal too close to its source, and cannot be used directly for aeroacoustic simulations.)

The second option is considered here as it seems to require less work and makes use of the vast amount of experience accumulated in flow modeling. CFD codes have built-in capabilities of handling non-linearities, curved boundaries, boundary layers, turbulence, and thermal effects. They are based on optimized, efficient, readily converging algorithms. If no CFD code is used as a basis, all these features have to be implemented again in the new code developed for the simulation of the sound field.

2. The need for a special approach to sound

Aerodynamic sound is generated as a result of the interaction of vortex structures that arise in viscous flows. These vortex structures are most often associated with either a shear layer or a solid surface. Once the sound is generated it propagates in the surrounding non-uniformly moving medium and travels to the ‘far field’.

Sound propagation is hardly affected by viscosity (that is why noise is so difficult to suppress). Also, sound perturbations are so small that their contribution to the

1991 *Mathematics Subject Classification*. Primary 65M60; Secondary 76D05.

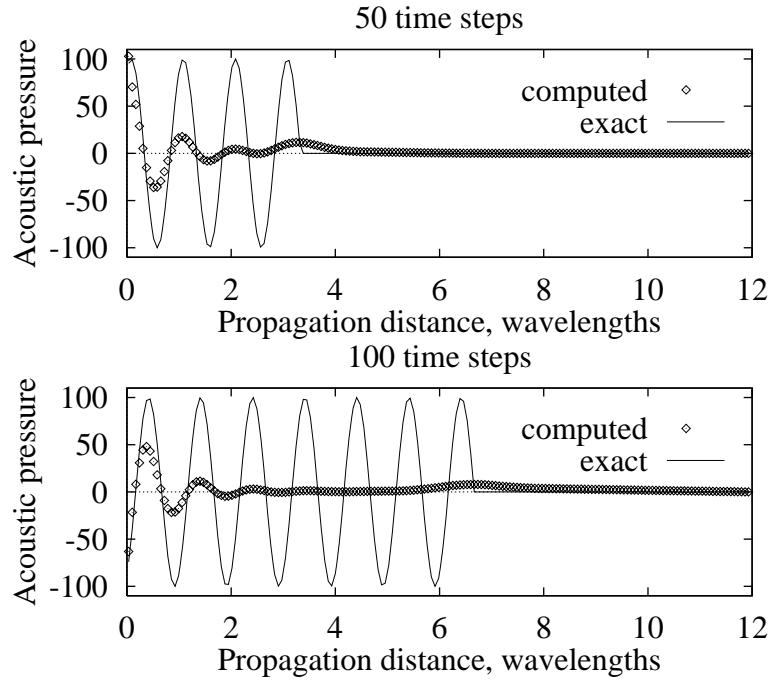


FIGURE 1. Conventional CFD solution of test problem

convection velocity of the flow is negligible in many cases. These two facts mean that sound propagation is, in essence, described by the linearized Euler equations (1).

The simulation of the flow that generates sound, however, requires time accurate solutions of the Navier-Stokes equations. Two approaches exist here: Reynolds Averages and Large Eddy Simulation. Both of them require adequate turbulence models and fine meshes to capture the small structures in the flow that oscillate and generate sound.

Most available Computational Fluid Dynamics (CFD) codes have implementations of Reynolds Averaged Navier-Stokes solvers (RANS). That is because the new alternative, Large Eddy Simulation (LES), requires more computational power that has become available only in the recent years. In our opinion, the future of Computational Aeroacoustics (CAA) is closely related to LES. For the time being, however, we should try to make the most of RANS.

Due to the diffusivity of the numerical schemes and the extremely small magnitude of the sound perturbations, RANS codes are not generally configured to simulate sound wave propagation. This is illustrated by the simple test of one-dimensional propagation in a tube of sound waves generated by a piston at one end that starts oscillating at time zero. The resulting sound field (pressure distribution) is compared with the one computed by the RANS solver PHOENICS [1] with its default numerical scheme (upwind fully implicit). As it can be seen on Figure 1, the numerical and the analytic solutions agree only in a very narrow region next to the source at the left end of the domain. In this admittedly worst-case scenario,

TABLE 1

<u>CFD</u> (Computational Fluid Dynamics)	<u>CAA</u> (Computational Aeroacoustics)
Nonuniform/Unstructured Grid	Regular Cartesian Grid
Fully Implicit in Time	Explicit/semi-implicit Schemes
Upwind Discretization	Higher Order Numerical Schemes
Smooth Solid Boundaries	Boundaries Can Be Stepwise
Small-scale structures	Extremely small magnitude

refining of the mesh does not change the result at all. (Better results can be obtained by switching to higher order schemes available within the same CFD code but they still cannot be relied upon for long distance wave propagation.)

To tackle these problems the new scientific discipline Computational Aeroacoustics has emerged in the last several years. The important issues of sound simulation have been identified [7], and adequate methods have been developed [8, 4, 3]. Table 1 shows how different the requirements for *accuracy* and *efficiency* are with the numerical solutions of the flow and the sound field respectively.

Although the sound equations (1) are a particular form of the equations governing fluid flow, great differences exist in magnitude, energy and scale of the solved-for quantities. (Acoustic perturbations are typically at least 10 times weaker than the corresponding hydrodynamic perturbations and a thousand times smaller than the mean flow that carries them. On the other hand acoustic wavelengths are typically several times larger than the corresponding structures in the flow.)

All this means that the algorithmic implementations are so different that they can hardly share any software modules. So, it will be best if a way is found of coupling a flow solver with an acoustic solver in such a manner that each of them does the job that it is best suited for.

3. The coupling

The basic idea of software coupling between CFD and CAA (decomposition of variables into flow and acoustic parts) as well as the Domain Decomposition into near field and far field was presented in our previous works [2, 3]. The CFD code is used to solve the time-dependent RANS equations while the CAA deals with the linearized Euler equations:

$$(1) \quad \begin{aligned} \frac{\partial p}{\partial t} + \bar{v}_j \frac{\partial p}{\partial x_j} + \bar{\rho} c^2 \frac{\partial v_j}{\partial x_j} &= S \\ \frac{\partial v_i}{\partial t} + \bar{v}_j \frac{\partial v_i}{\partial x_j} + \frac{1}{\bar{\rho}} \frac{\partial p}{\partial x_i} &= F_i. \end{aligned}$$

Here p is the pressure perturbation, v_1, v_2 and v_3 are the Cartesian components of the velocity perturbation. The values of the speed of sound c , of the local density $\bar{\rho}$ and of the velocity components of the flow \bar{v}_j are supplied by the CFD code.

CAA algorithms are designed to solve these equations (1) with known right-hand sides S and F_i that are functions of x_i and time t . Term S contains any sources of mass that may be present in the computational domain, such as vibrating solid surfaces. The three forcing terms F_i will be set to zero in most practical acoustic applications. In theory they contain the viscous forces which have negligible effect on sound propagation. There are some cases where the nonlinear terms associated with the acoustic perturbations may have to be taken into account. Then S and F_i will be updated within the acoustic code at each iteration rather than once per time step.

The present study concentrates on the use of the source term S to transfer the information about the generation of sound from the CFD code to the acoustic solver.

A closer examination of the time history of the CFD solution pictured in Figure 1 reveals that the pressure at the first node next to the source of sound has been resolved accurately. It is suggested that the temporal derivative of the local pressure at the source nodes, calculated from the CFD solution, is added to the source term S of the acoustic equations (1).

$$(2) \quad S = \frac{\partial \bar{p}}{\partial t} + S_{vib}$$

Here S_{vib} denotes sources external to the flow like vibrating solid objects. Thus the following combined algorithm can be outlined:

1. Obtain a steady CFD solution of the flow problem.
2. Start the time-dependent CFD simulation with these initial conditions.
3. Impose the calculated temporal derivative of the pressure at selected nodes within the flow region as part of the source term of the acoustic simulation.
4. Solve the linearized Euler equations in the acoustic domain applying any external sources of mass (vibrating solids).

The introduction of the temporal derivative $\frac{\partial \bar{p}}{\partial t}$ into the source term S is best implemented in finite volume discretization. Then, if phase accuracy of the calculated acoustic signal is essential (like with resonance), the time-dependent outflow from the control volume with increasing \bar{p} in the CFD solution or the inflow perturbation if \bar{p} is decreasing should also be taken into account in order to calculate the correct amount of mass that is assumed to enter or exit the acoustic simulation at each time step. The finite volume form of the RANS continuity equation with isentropic conditions ($\frac{\partial \bar{p}}{\partial p} = c^2$) suggests the following pressure source:

$$(3) \quad \begin{aligned} S &= \bar{\rho}c^2(\bar{v}_j - \bar{v}_{j,average})\frac{A_j}{\Delta V} + S_{vib} \\ j &= \text{Influx}, \frac{\partial \bar{p}}{\partial t} > 0 \\ j &= \text{Outflow}, \frac{\partial \bar{p}}{\partial t} < 0 \end{aligned}$$

where A_{Influx} is the area of the faces of the cell with volume ΔV across which there is inflow during the time step Δt , and the repeated index denotes summation over all such faces. Since this option introduces the whole flow perturbation into

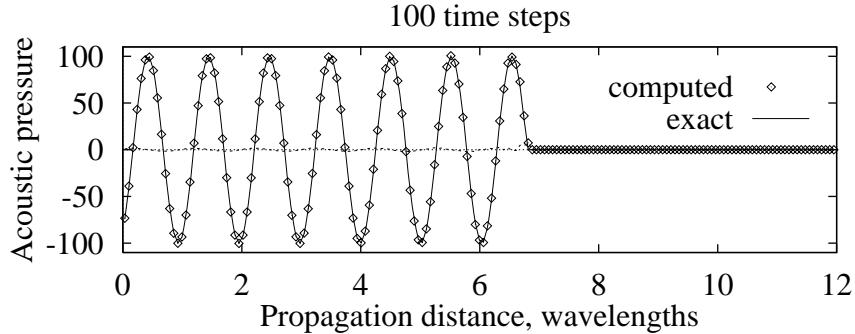


FIGURE 2. Combined solution of test problem

the acoustic simulation, the acoustic solver in this case must be capable of handling smooth curved solid boundaries.

The two codes have separate meshes in overlapping domains. The RANS mesh must be body fitted to represent smooth solid boundaries. The acoustic mesh can be regular Cartesian if option (2) is chosen. In this case the CAA domain can be larger — extending to the mid field if Kirchhoff's method is used [5] or to the far field if high-order optimized numerical schemes are employed [8]. Uniform mean flow has to be assumed outside the region of the CFD simulation.

Prior to the introduction into the acoustic simulation the flow quantities (\bar{v}_j , $\bar{\rho}$ and \bar{p} , in air $c^2 = 1.4\frac{\bar{p}}{\bar{\rho}}$, see 1) are interpolated with piecewise constant functions (choosing the nearest neighbouring point from the irregular CFD mesh). This can be done because typically the flow mesh is finer than the acoustic mesh.

In some cases (separated flows, jets) the sources of noise cannot be localized and are instead distributed across the computational domain. Then the most efficient option is choosing a higher order scheme within the CFD code itself and defining a ‘near field’ boundary where the acoustic signal is radiated from the RANS solver to the linearized Euler solver.

4. Results

The above algorithm was validated on the same 1D propagation test problem that was described in the second section. Using the pressure, velocity and density fields provided by the CFD code at each time step and a finite volume acoustic solver, the actual acoustic signal was recovered as it can be seen in Figure 2. Here the coupling option defined by (3) has been implemented with $S_{vib} = 0$. (The acoustic source was introduced in the CFD simulation rather than in the acoustic one, in order to set up this test.)

As a 2D example, generation of sound by a vortex impinging on a flat plate is considered. A Reynolds-Averaged Navier-Stokes solver [1] is used to compute the airflow on a mesh that is two times finer in the direction perpendicular to the plate than the corresponding grid for adequate acoustic simulation.

The geometry of the problem and the hydrodynamic perturbation velocity field (with the uniform background flow subtracted) are in Figure 3.

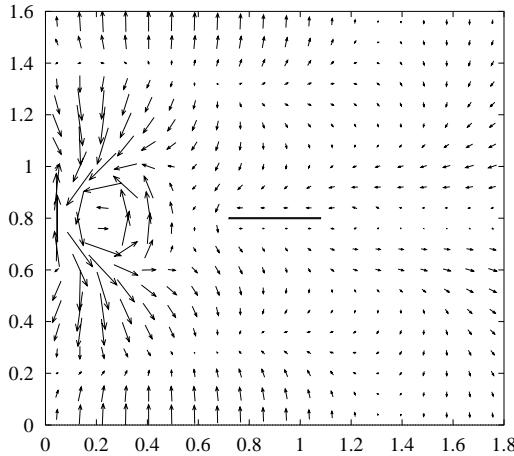


FIGURE 3. Hydrodynamic perturbations and blade. Scale: 1 m/s to 0.1 m

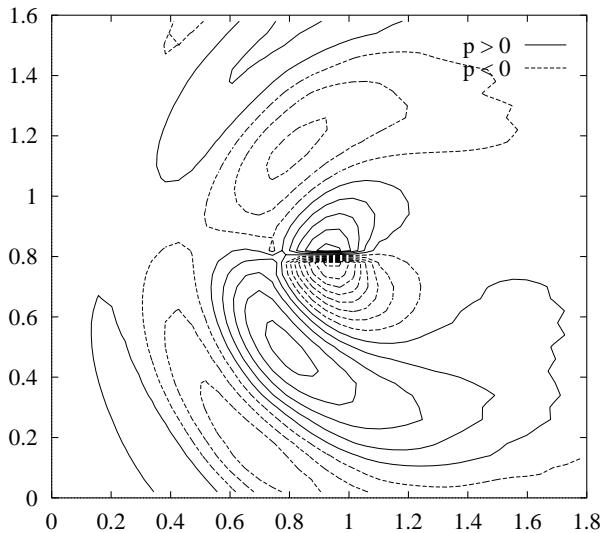


FIGURE 4. Instantaneous pressure contours. Spacing: 6 Pa

The pressure fluctuations (temporal derivatives) next to the solid surface are imposed as source terms on the linearised Euler equations which are solved separately as described above. The size of the computational domain is small enough so that the finite volume solver [3] can predict accurately the sound field. A snapshot of the pressure perturbations can be seen in Figure 4. A graph was made of the acoustic pressure as a function of time at different locations above and below the solid blade. As it can be seen from Figure 5, the amplitude of the sound waves generated at the blade decreases away from it as expected.

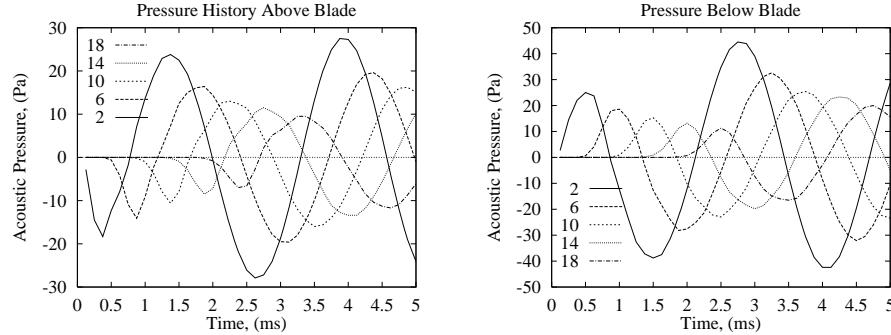


FIGURE 5. Acoustic signal in the specified cells above and below the centre of the blade.

5. Conclusions

A coupled technique has been developed that allows general-purpose RANS solvers to be used with sound generation problems. Both geometrical and physical domain decomposition has been considered. At the locations where it is generated sound is passed to a linearized Euler solver that allows adequate numerical representation of the propagating acoustic waves. The current implementation is applicable to aerodynamic noise generation either on solid surfaces or in volumes that are not surrounded by reflecting objects.

References

1. CHAM Ltd, Wimbledon, UK, *Phoenics*.
2. G.S. Djambazov, C.-H. Lai, and K.A. Pericleous, *Development of a domain decomposition method for computational aeroacoustics*, DD9 Proceedings, John Wiley & Sons, 1997.
3. ———, *Domain decomposition methods for some aerodynamic noise problems*, 3rd AIAA/CEAS Aeroacoustics Conference, no. 97-1608, 1997, pp. 191–198.
4. ———, *Testing a linear propagation module on some acoustic scattering problems*, Second Computational Aeroacoustics Workshop on Benchmark Problems, Conference Publications, no. 3352, NASA, 1997, pp. 221–229.
5. Anastasios S. Lyrintzis, *The use of Kirchhoff's method in computational aeroacoustics*, ASME-FED **147** (1993), 53–61.
6. A.D. Pierce, *Validation methodology: Review and comments*, Computational Aeroacoustics, Springer-Verlag New York, Inc., 1993, pp. 169–173.
7. C.K.W. Tam, *Computational aeroacoustics: Issues and methods*, AIAA Journal **33** (1995), no. 10, 1788–1796.
8. C.K.W. Tam and J.C. Webb, *Dispersion-relation-preserving finite difference schemes for computational acoustics*, Journal of Computational Physics **107** (1993), 262–281.

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH, WELINGTON STREET, WOOLWICH, LONDON SE18 6PF, U.K.

E-mail address: G.Djambazov@gre.ac.uk

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH, WELINGTON STREET, WOOLWICH, LONDON SE18 6PF, U.K.

E-mail address: C.H.Lai@gre.ac.uk

SCHOOL OF COMPUTING AND MATHEMATICAL SCIENCES, UNIVERSITY OF GREENWICH, WELINGTON STREET, WOOLWICH, LONDON SE18 6PF, U.K.

E-mail address: K.Pericleous@gre.ac.uk

Non-overlapping Domain Decomposition Applied to Incompressible Flow Problems

Frank-Christian Otto and Gert Lube

1. Introduction

A non-overlapping domain decomposition method with Robin-type transmission conditions which is known for scalar advection-diffusion-reaction problems [2],[5] is generalized to cover the Oseen equations. The presented method, which is later referred to as DDM, is an additive iteration-by-subdomains algorithm. Hence parallelism is given in a very natural way. The formulation is based on the continuous level to study the DDM without dealing with a special discretization. A convergence result for the “continuous” algorithm is presented. To treat incompressible Navier-Stokes problems, the

A parallel implementation based on a finite element discretization has been done. Numerical results indicating linear convergence with a rate independent of the mesh size are presented for both the (linear) Oseen equations and the (non-linear) Navier-Stokes equations.

We denote by $L^2(\Omega)$ the space of square integrable functions with norm $\|\cdot\|_{0,\Omega}$ and inner product $(\cdot, \cdot)_\Omega$. $H^s(\Omega)$ denotes the usual Sobolev space with norm $\|\cdot\|_{s,\Omega}$. For $\Gamma \subset \partial\Omega$ we write $\langle \cdot, \cdot \rangle_\Gamma$ for the inner product in $L^2(\Gamma)$ (or, if needed, for the duality product between $H_{00}^{\frac{1}{2}}(\Gamma)$ and $H_{00}^{-\frac{1}{2}}(\Gamma)$). The space $H_{00}^{\frac{1}{2}}(\Gamma)$ consists of functions $u \in H^{\frac{1}{2}}(\Gamma)$ with $d^{-\frac{1}{2}}u \in L^2(\Gamma)$ where $d(x) = \text{dist}(x, \partial\Gamma)$ [3, Chap 1., Sec. 11.4]. We explain the DDM for the Oseen equations in Section 2 and look into its analysis in Section 3. Then we explain how to discretize the method (Section 4) and apply it to the Navier-Stokes equations (Section 5). Numerical results are presented in Section 6.

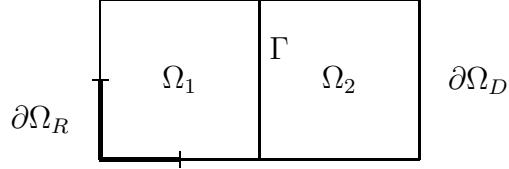
2. Definition of the DDM for the Oseen equations

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with Lipschitz, piecewise C^2 -boundary $\partial\Omega$. We consider the following boundary value problem for the Oseen equations

$$(1) \quad \left\{ \begin{array}{rcl} -\nu \Delta \mathbf{u} + \nabla p + (\mathbf{b} \cdot \nabla) \mathbf{u} + c \mathbf{u} & = & \mathbf{f} \in (L^2(\Omega))^2 \\ \nabla \cdot \mathbf{u} & = & 0 \in L^2(\Omega) \\ \mathbf{u} & = & \mathbf{g} \in (H^{\frac{1}{2}}(\partial\Omega_D))^2 \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} + \eta \mathbf{u} & = & \mathbf{h} \in (L^2(\partial\Omega_R))^2 \end{array} \right. ,$$

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 76C99.

FIGURE 1.



where $\partial\Omega = \overline{\partial\Omega}_D \cup \overline{\partial\Omega}_R$, $\partial\Omega_D \cap \partial\Omega_R = \emptyset$, and $\nu > 0$, $\mathbf{b} \in (H^1(\Omega))^2$ with $\nabla \cdot \mathbf{b} \in L^\infty(\Omega)$, $c \in L^\infty(\Omega)$, $\eta \in L^\infty(\partial\Omega_R)$. \mathbf{n} is the outer normal on $\partial\Omega$. If $\partial\Omega_R = \emptyset$ we also require $\int_{\partial\Omega} \mathbf{g} \cdot \mathbf{n} = 0$. Then if $c - \frac{1}{2}\nabla \cdot \mathbf{b} \geq 0$ and $\eta + \frac{1}{2}\mathbf{b} \cdot \mathbf{n} \geq \eta_0 = \text{const} > 0$ problem (1) has a unique solution which belongs to $(H^1(\Omega))^2 \times L^2(\Omega)$ if $\mu(\partial\Omega_R) > 0$, and to $(H^1(\Omega))^2 \times L_0^2(\Omega)$ with $L_0^2(\Omega) = \{f \in L^2(\Omega) \mid \int_D f dx = 0\}$ otherwise.

The reason why we consider a mixed boundary value problem will become clear in the next section. Here we allow the additional term $c\mathbf{u}$ within the momentum equation which occurs if a simultaneous linearization and semi-discretization in time of the non-stationary Navier-Stokes equations is performed.

A heuristical approach to non-overlapping domain decomposition methods for this type of problems is as follows. We divide Ω into two subdomains Ω_k , $k = 1, 2$ also having a Lipschitz, piecewise C^2 -boundary. The artificial boundary $\partial\Omega_1 \cap \partial\Omega_2$ is denoted by Γ (Figure 1). For simplicity we assume $\overline{\partial\Omega_R} \subsetneq \partial\Omega_1 \cap \partial\Omega$.

Then the original boundary value problem is equivalent to the following split formulation (\mathbf{n}_i always denotes the outer normal of Ω_i)

$$(2) \left\{ \begin{array}{lcl} -\nu\Delta\mathbf{u}_1 + \nabla p_1 + (\mathbf{b} \cdot \nabla)\mathbf{u}_1 + c\mathbf{u}_1 & = & \mathbf{f} \in (L^2(\Omega_1))^2 \\ \nabla \cdot \mathbf{u}_1 & = & 0 \in L^2(\Omega_1) \\ \mathbf{u}_1 & = & \mathbf{g} \in (H^{\frac{1}{2}}(\partial\Omega_D)|_{\partial\Omega_D \cap \partial\Omega_1})^2 \\ \nu \frac{\partial\mathbf{u}_1}{\partial\mathbf{n}_1} - p_1 \mathbf{n}_1 + \eta \mathbf{u}_1 & = & \mathbf{h} \in (L^2(\partial\Omega_R))^2 \end{array} \right.$$

$$(3) \left\{ \begin{array}{lcl} -\nu\Delta\mathbf{u}_2 + \nabla p_2 + (\mathbf{b} \cdot \nabla)\mathbf{u}_2 + c\mathbf{u}_2 & = & \mathbf{f} \in (L^2(\Omega_2))^2 \\ \nabla \cdot \mathbf{u}_2 & = & 0 \in L^2(\Omega_2) \\ \mathbf{u}_2 & = & \mathbf{g} \in (H^{\frac{1}{2}}(\partial\Omega_D)|_{\partial\Omega_D \cap \partial\Omega_2})^2 \end{array} \right.$$

together with the continuity requirements on Γ

$$(4) \quad \mathbf{u}_1 = \mathbf{u}_2 \quad \text{in } (H^{\frac{1}{2}}(\Gamma))^2,$$

$$(5) \quad \nu \frac{\partial\mathbf{u}_1}{\partial\mathbf{n}_1} - p_1 \mathbf{n}_1 = -\nu \frac{\partial\mathbf{u}_2}{\partial\mathbf{n}_2} + p_2 \mathbf{n}_2 \quad \text{in } (H^{-\frac{1}{2}}(00))^2.$$

These two continuity conditions can be used to construct a non-overlapping domain decomposition method for this problem, which can be considered as an iterative decoupling of the split formulation. For example using (4) iteratively to calculate solutions on Ω_1 , i.e. $\mathbf{u}_1^k = \mathbf{u}_2^{k-1}$, and (5) for Ω_2 we would get a Dirichlet-Neumann-algorithm [7].

We use however a linear combination of both conditions for all subdomains, i.e. to get a new solution on Ω_i within the iteration, we impose

$$(6) \quad \nu \frac{\partial\mathbf{u}_i^k}{\partial\mathbf{n}_i} - p_i^k \mathbf{n}_i + \lambda_i \mathbf{u}_i^k = \nu \frac{\partial\mathbf{u}_j^{k-1}}{\partial\mathbf{n}_i} - p_j^{k-1} \mathbf{n}_i + \lambda_i \mathbf{u}_j^{k-1} \quad \text{on } \partial\Omega_i \cap \partial\Omega_j$$

Such an approach does not require a red-black partition which is often needed for DN-algorithms. Furthermore it is known for the related method for advection-diffusion-reaction problems that the behaviour of the solution can be modelled “adaptively” if λ_i is chosen as an appropriate function [2],[5],[1],[6].

More precisely we use a restricted class of weighting factors λ_i as given below, but we allow some kind of relaxation of the interface condition (6).

The algorithm. We now consider a non-overlapping partition into N subdomains $\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i$, $\Omega_i \cap \Omega_j = \emptyset \forall i \neq j$ with each Ω_i having the same boundary regularity as Ω . We denote $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$ and $\bar{\Gamma}_{ij} = \partial\Omega_i \cap \partial\Omega_j$.

Furthermore we use the following notations:

$$\text{“interface operator”} : \Phi_i(\mathbf{u}, p) = \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}_i} - p \mathbf{n}_i + (-\frac{1}{2} \mathbf{b} \cdot \mathbf{n}_i + \rho_i) \mathbf{u}$$

$$\text{initial interface condition} : \Phi_{i,0}$$

$$\text{“relaxation parameter”} : \theta \in (0, 1]$$

Instead of λ_i we use $-\frac{1}{2} \mathbf{b} \cdot \mathbf{n}_i + \rho_i$ with ρ_i to be chosen, because the necessary restrictions are easier formulated for ρ_i .

Now the domain decomposition algorithm for the Oseen problem (1) reads:

For $k \in \mathbb{N}$ solve for all subdomains Ω_i ($i = 1, \dots, N$) in parallel:

$$(7) \quad \begin{cases} -\nu \Delta \mathbf{u}_i + \nabla p_i + (\mathbf{b} \cdot \nabla) \mathbf{u}_i + c \mathbf{u}_i &= \mathbf{f} \\ \nabla \cdot \mathbf{u}_i &= 0 \end{cases}$$

with the given boundary conditions on $\partial\Omega_i \cap \partial\Omega$ together with the interface condition

$$(8) \quad \Phi_i(\mathbf{u}_i^k, p_i^k) = \begin{cases} \theta \Phi_i(\mathbf{u}_j^{k-1}, p_j^{k-1}) + (1 - \theta) \Phi_i(\mathbf{u}_i^{k-1}, p_i^{k-1}) & k > 1 \\ \Phi_{i,0} & k = 1 \end{cases}$$

on Γ_{ij} .

3. Convergence Analysis

Before formulating the convergence result for the algorithm above we start with its well-posedness.

LEMMA 1. *In addition to the regularity of the data prescribed in Section 2 we assume $c - \frac{1}{2} \nabla \cdot \mathbf{b} \geq 0$, $\eta + \frac{1}{2} \mathbf{b} \cdot \mathbf{n} \geq \eta_0 = \text{const} > 0$, and for all subdomains Ω_i*

1. $\rho_i \in L^\infty(\Gamma_i)$ with $\rho_i \geq \rho_i^0 = \text{const} > 0$
2. $\Phi_{i,0} \in L^2(\Gamma_i)$
3. $c - \frac{1}{2} \nabla \cdot \mathbf{b} \geq c_i = \text{const} > 0$ or $\mu(\partial\Omega_i \cap \partial\Omega) > 0$.

Then the domain decomposition algorithm is well-defined, i.e. all local boundary value problems have for all k a unique solution in $(H^1(\Omega_i))^2 \times L^2(\Omega_i)$. Furthermore we have

$$(9) \quad \Phi_i(\mathbf{u}_i^k, p_i^k) \in L^2(\Gamma_{ij}) \quad \forall k.$$

We emphasize that the local pressure solution p_i^k is unique in $L^2(\Gamma_i)$ for all k .

Now we denote by Π_i the L^2 -projection onto the space $L_0^2(\Omega_i)$, more precisely

$$(10) \quad \Pi_i : L^2(\Omega_i) \rightarrow L_0^2(\Omega_i) \quad q \mapsto q - \frac{1}{\mu(\Omega_i)} \int_{\Omega_i} q \, dx.$$

THEOREM 2. *Let the solution (\mathbf{u}, p) of (1) be regular enough to have $\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}_i} - p \mathbf{n}_i \in (L^2(\Gamma_{ij}))^2$ for all i, j . If $\rho_i = \rho_j$ a.e. then we have under the assumptions of Lemma 1 for all $\theta \in (0, 1]$*

$$\begin{aligned}\|\mathbf{u}_i^k - \mathbf{u}_i\|_{1,\Omega_i} &\longrightarrow 0, \\ \|\Pi_i(p_i^k - p_i)\|_{0,\Omega_i} &\longrightarrow 0\end{aligned}$$

for $k \rightarrow \infty$, where (\mathbf{u}_i, p_i) is the restriction of (\mathbf{u}, p) to Ω_i .

Furthermore, if $\mu(\partial\Omega_R) > 0$, i.e. a mixed boundary value problem is considered, we have for all $\theta \in (0, 1]$

$$\|p^k - p\|_{0,\Omega_i} \longrightarrow 0$$

for $k \rightarrow \infty$.

Remarks

- If $c(x) - \frac{1}{2}\nabla \cdot \mathbf{b}(x) \geq C > 0$ then arbitrary subdomain partitions satisfying the regularity requirements are allowed. Especially internal cross-points can be treated. For $C = 0$ such partitions are not covered by the theorem, but nevertheless they work in numerical computations.
- The Stokes problem is covered.
- All results remain valid if different ρ_i for every velocity component are used.
- For a Dirichlet problem the pressure convergence is local: Only the locally normalized pressure will converge. I.e. we have pressure convergence up to a constant which can differ for different subdomains and iteration steps.
- Since the theorem yields no information about the convergence speed, we have no theoretical indication how to construct a “good” ρ_i . A heuristic approach to a “good” ρ_i for advection-diffusion-reaction equations is contained in [6]. For the Oseen equations this question is still open.

To prove the convergence for the velocity variable a key step is the relation

$$\begin{aligned}\|\mathbf{u}_i^k\|_i^2 + \int_{\Gamma_i} \frac{1}{4\rho_i} (\Phi_i(\mathbf{u}_i^k, p_i^k) - 2\rho_i \mathbf{u}_i^k)^2 &= \int_{\Gamma_i} \frac{1}{4\rho_i} (\Phi_i(\mathbf{u}_i^k, p_i^k))^2 \\ \text{with } \|\mathbf{u}\|_i^2 := \nu |u|_{1,\Omega_i}^2 + \|(c - \frac{1}{2}\nabla \cdot \mathbf{b})^{\frac{1}{2}} \mathbf{u}\|_{0,\Omega_i}^2 + \|(\eta + \frac{1}{2}\nabla \cdot \mathbf{b})^{\frac{1}{2}} \mathbf{u}\|_{0,\partial\Omega_R \cap \partial\Omega_i}^2\end{aligned}$$

which uses the L^2 -regularity of the interface data. This part is established similar to the convergence of the related algorithm for advection-diffusion-reaction problems. ([1] contains that proof for $\theta = 1$.)

The local pressure convergence comes from a modified a priori estimate which is based on the continuous version of the Babuška-Brezzi-condition. Global convergence of p in the case $\mu(\partial\Omega_R) > 0$ is based on the transmission of the local pressure mean values across interfaces. The full proof is given in [6].

4. The discrete algorithm

Since finite elements are favoured as discretization method, weak formulations of the subdomain problems should be considered. In the case of homogeneous Dirichlet boundary conditions on $\partial\Omega$ and $c = 0$ the local Oseen problems read in weak formulation:

Within step k find $(\mathbf{u}_i^k, p_i^k) \in V_i \times Q_i = \{\mathbf{v} \in (H^1(\Omega_i))^2 \mid \mathbf{v} = 0 \text{ on } \partial\Omega \cap \partial\Omega_i\} \times L^2(\Omega_i)$ with

$$(11) \quad \begin{aligned} a_i(\mathbf{b}; \mathbf{u}_i^k, \mathbf{v}) - b_i(p_i^k, \mathbf{v}) + b_i(q, \mathbf{u}_i^k) &+ \langle (-\frac{1}{2}\mathbf{b} \cdot \mathbf{n}_i + \rho_i)\mathbf{u}_i^k, \mathbf{v} \rangle_{\Gamma_i} \\ &= (f, \mathbf{v})_{\Omega_i} + \sum_{j \neq i} \langle \Lambda_{ji}^{k-1}, \mathbf{v} \rangle_{\Gamma_{ij}} \end{aligned}$$

where

$$\begin{aligned} a_i(\mathbf{b}; \mathbf{u}, \mathbf{v}) &= \nu(\nabla \mathbf{u}, \nabla \mathbf{v})_{\Omega_i} + (\mathbf{b} \cdot \nabla \mathbf{u} + c\mathbf{u}, \mathbf{v})_{\Omega_i} \\ b_i(q, \mathbf{v}) &= (p, \nabla \cdot \mathbf{v})_{\Omega_i} \\ \Lambda_{ji}^k &= \theta \Phi_i(\mathbf{u}_j^k, p_j^k) + (1 - \theta) \Phi_i(\mathbf{u}_i^k, p_i^k). \end{aligned}$$

The discretization is performed by choosing finite dimensional subspaces V_i^h, Q_i^h of V_i, Q_i which consist of piecewise polynomial functions on the restriction of a global triangulation of Ω to Ω_i .

The evaluation of $\Phi_i(\mathbf{u}_j^{k-1}, p_j^{k-1})$ resp. $\Phi_i(\mathbf{u}_i^{k-1}, p_i^{k-1})$ can be avoided by means of the following formula

$$(12) \quad \Lambda_{ij}^k = \theta(\rho_i + \rho_j)\mathbf{u}_i^k - \theta\Lambda_{ji}^{k-1} + (1 - \theta)\Lambda_{ij}^{k-1}$$

which does not use derivatives of the finite element solutions. Again the discrete algorithm starts with an initial guess $\Phi_{i,0}$ for the interface condition. Hence a good initial guess can reduce the number of iterations until convergence.

5. Application to the Navier-Stokes equations

The stationary Navier-Stokes equations as a non-linear problem of the form

$$A[\hat{u}]\hat{u} = f$$

can be solved by a defect correction method

$$(13) \quad A[\hat{u}^{m-1}](\hat{u}^m - \hat{u}^{m-1}) = \omega_m \{f - A\hat{u}^{m-1}\}$$

or equivalently

$$(14) \quad A[\hat{u}^{m-1}](\hat{u}^m) = \omega_m f + (1 - \omega_m)A\hat{u}^{m-1}$$

with some damping factor $\omega_m > 0$. The idea is to solve the linear(ized) Oseen problem occurring within this iterative process using the domain decomposition algorithm as inner cycle. Then the local subproblems are as in (11) with \mathbf{u}_i^k, p_i^k replaced by $\mathbf{u}_i^{m,k}, p_i^{m,k}$ and \mathbf{b} by the velocity solution from the previous linearization step. If the formulation (14) is used, an appropriate initial interface condition for the domain decomposition within step m is the last calculated Λ_{ij}^k from step $m-1$. Hence results of step $m-1$ are re-used and it is not necessary to achieve convergence of the domain decomposition algorithm within every linearization step.

6. Numerical examples

As remarked in Section 2 an analogous method turned out to be very efficient for advection-diffusion-reaction problems [1], [6]. Hence for the numerical examples below we used the straightforward extension of the interface function proposed in [1]

$$(15) \quad \rho_i = \sqrt{(\mathbf{b} \cdot \mathbf{n}_i)^2 + \nu \lambda}.$$

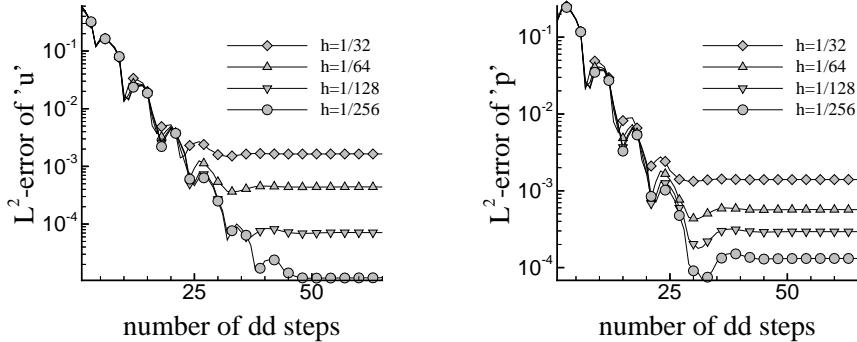


FIGURE 2. Convergence history for Example 1.

λ is a strictly positive function and could be chosen separately for each velocity component.

Within our calculations we use continuous piecewise linear finite elements for the velocity as well as for the pressure. So we add to (11) residual terms in order to satisfy a modified Babuška-Brezzi-condition and to get a stable discretization (see [4] for details).

Numerical experiments showed that a relaxation parameter $\theta < 1$ gives global pressure convergence for the Dirichlet problem, too (cf. Theorem 2). But for the type of problems considered below there is no acceleration for smaller θ . So we chose $\theta = 1$ for all test cases.

Example 1. Linearized Navier-Stokes flow (Oseen flow):

We consider the Poiseuille flow in a 2d channel, where we use the quadratic profile as known velocity field. At the outflow part we impose a homogeneous Neumann boundary condition and prescribe the velocities elsewhere. The computational domain $[0, 1] \times [0, 1/4]$ is divided into 4 subdomains arranged in a row. The exact solution is given by $(u, v, p) = (64y(1/4 - y), 0, -128\nu x)$ with $\nu = 10^{-3}$. We show in Figure 2 the convergence history of the discrete L^2 -errors versus the iteration number of the DDM for different mesh sizes. (Due to their interface discontinuities the dd-solutions do not belong to the space of continuous finite element functions which is needed to calculate residuals directly. That is why we here only consider the error. An alternative is under development.)

The results indicate that the DDM converges almost linearly until a certain error level is achieved which corresponds to the mesh size. The rate of convergence seems to be independent of the mesh size. In comparison to the related algorithm for scalar equations [6] the performance is worse and the choice of λ is more critical. Here it is chosen as $\lambda = 5/\nu$.

So far no mechanism of global data transport (like a coarse grid) is incorporated in the algorithm; hence it cannot be scalable. Table 1 shows the dependence of the number of subdomains for the finest grid used for this example. Neither load-balancing nor inexact subdomain solving has been used to obtain these results. As expected the number of iterations increases with the number of subdomains. Nevertheless, the computing time decreases and this suggests that this algorithm together

TABLE 1. Iteration numbers and computing time needed on the finest mesh ($h = 1/256$) to achieve for Example 1. an u -error smaller than $5 \cdot 10^{-5}$.

subdomain partition	number of dd-iterations	CPU-time [s] (fastest and slowest subdomain)
2×1	17	1260, 1630
4×1	54	1000, 1410
4×2	88	540, 1110



FIGURE 3. Subdomain partition for Example 2.

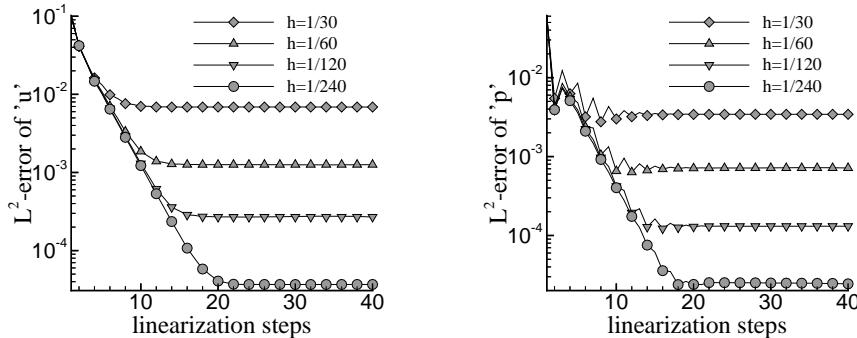


FIGURE 4. Convergence history for Example 2.

with a coarse grid solver can be very efficient for larger numbers of subdomains.

Example 2. Stationary Navier-Stokes flow around a cylinder:
We consider the stationary flow in a 2d-channel with an obstacle. We have a quadratic profile at the inflow, no-slip conditions at the walls and a homogeneous Neumann boundary condition at the outflow. The viscosity is $\nu = 10^{-3}$ and we choose $\lambda = 1/\nu$ in (15). The computational domain has been divided into 8 subdomains as shown in Figure 3.

In Figure 4 we show the convergence history of the discrete L^2 -errors versus the number of linearization steps for different mesh sizes. Within each step we performed 10 steps of the domain decomposition algorithm. To calculate the errors we used a reference solution obtained by solving the global boundary value problem on the same mesh up to the level of the truncation error.

The graphs show the linear convergence of the outer iteration (linearization) with a rate independent of the mesh size. A direct computation without the DDM needs between 13 and 16 linearization steps. Hence with 10 dd steps within every linearization step we get nearly the same accuracy with roughly the same number of steps. In fact, the parallel calculation is cheaper with respect to computing time. So the DDM works quite well as kernel of a Navier-Stokes solver.

7. Summary

We described a non-overlapping domain decomposition algorithm for the Oseen (linearized Navier-Stokes) equations and proved its convergence on the continuous level. A discretized variant was proposed and applied to the Navier-Stokes problem. A finite element implementation which has not been fully optimized yielded reasonable results for the linear and non-linear problem. The method has also been applied to non-isothermal flow problems with promising results. Further investigations of both theory and implementation are under development.

References

1. A. Auge, A. Kapurkin, G. Lube, and F.-C. Otto, *A note on domain decomposition of singularly perturbed elliptic problems*, in: Proceedings of the Ninth International Conference on Domain Decomposition, Bergen, 1996 (in press).
2. C. Carlenzoli and A. Quarteroni, *Adaptive domain decompositon methods for advection-diffusion problems*, Modeling, Mesh Generation, and Adaptive Numerical Method for Partial Differential Equations (I. Babuška et al., eds.), Institute for Mathematics and its Applications IMA Volume 75, Springer Verlag, New York a.o., 1995.
3. J.-L. Lions and E. Magenes, *Non-homogeneous boundary value problems and applications I*, Springer-Verlag, 1972.
4. G. Lube, *Stabilized Galerkin finite element methods for convection dominated and incompressible flow problems*, Numerical Analysis and Mathematical Modelling, Banach Center Publications, Volume 29, Polish Academy of Sciences, Warszawa, 1994, pp. 85–104.
5. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Mathematical Models and Methods in Applied Sciences 5 (1995), 67–93.
6. F.-C. Otto and G. Lube, *A non-overlapping domain decomposition method for the Oseen equations*, to appear in Mathematical Models and Methods in Applied Sciences.
7. A. Quarteroni, *Domain decomposition and parallel processing for the numerical solution of partial differential equations*, Surv. Math. Ind. 1 (1991), 75–118.

INSTIUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK, UNIVERSITÄT GÖTTINGEN, LOTZESTR. 16-18, 37083 GÖTTINGEN, GERMANY

E-mail address: otto@math.uni-goettingen.de

INSTIUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK, UNIVERSITÄT GÖTTINGEN, LOTZESTR. 16-18, 37083 GÖTTINGEN, GERMANY

E-mail address: lube@math.uni-goettingen.de

A Domain Decomposition Based Algorithm For Non-linear 2D Inverse Heat Conduction Problems

Charaka J. Palansuriya, Choi-Hong Lai, Constantinos S. Ierotheou,
and Koulis A. Pericleous

1. Introduction

Inverse heat conduction problems (IHCPs) appear in many important scientific and technological fields. Hence analysis, design, implementation and testing of inverse algorithms are also of great scientific and technological interest. The numerical simulation of 2-D and 3-D inverse (or even direct) problems involves a considerable amount of computation. Therefore, the investigation and exploitation of parallel properties of such algorithms are equally becoming very important [9, 2]. Domain decomposition (DD) methods are widely used to solve large scale engineering problems and to exploit their inherent ability for the solution of such problems.

An area of particular interest in IHCPs is the cutting of sheet material such as metal. An accurate simulation of the temperature distribution of the metal, subject to cutting, is vital in order to lengthen the life time of the cutting tool and to guarantee the quality of the cutting. In addition, the real-time simulation of such temperature distributions is of industrial interest. For example, it is important to regulate the cutter speed and coolant application in order to keep the temperature (especially at the cutter points) below a threshold. When the temperature rises above the threshold this will cause deformation of the metal or it may become fatigued. In reality, the accurate measurement of temperature at the cutter points is not possible. Therefore, a direct problem cannot be formulated. Inverse methods can be used to retrieve the temperature at these points. It has been shown that accurate estimates can be obtained using such methods [1]. IHCPs, such as the metal cutting problem described above, are more difficult to solve analytically than direct problems [1]. Therefore, various approximation methods have been developed to solve such problems. These include graphical[10], polynomial [5], Laplace transform [7], dynamic programming [11], finite difference [3], finite elements [6]

1991 *Mathematics Subject Classification*. Primary 65M55; Secondary 35K55, 65M06, 65Y05.

Key words and phrases. Domain Decomposition, Problem Partitioning, Metal Cutting Problems, Domain Parallelism, Domain-Data parallelism.

The first author is partially funded by the University of Greenwich.

Computing time on Origin 2000 was sponsored by Parallab at the University of Bergen.

and finite volume. Here we will use a finite volume based method. The main objective of this work is to study DD methods to solve IHCPs and to explore algorithms which are suitable for distributed/parallel computing environments.

This paper is organised as follows. First, a description of the mathematical model for the dimensionless 2D non-linear metal cutting problem is given. Second, the description of the problem partitioning is given. Different numerical schemes are used in different sub-domains in order to solve different sub-problems. Numerical results are shown for a metal cutting application. Third, the exploitation of the parallel properties of the numerical schemes are explained. The resulting parallel implementation uses MPI (Message Passing Interface) directives [4] and is suitable for network-cluster (distributed) computing as well as for traditional tightly-coupled multi-processor systems. Finally, some conclusions are drawn.

2. Dimensionless 2D Non-linear Metal Cutting Problems

The metal cutting problem considered here is a 2D thin sheet of metal defined in the domain $D = \{(x, y) : 0 < x < 1 \text{ and } 0 < y < 1\}$. The material property is assumed to be homogeneous across the domain of interest and the following assumptions are made for an idealised cutting :- (1) the application of a cutting tool at the cutter points is equivalent to the application of a source at these points, (2) no phase changes occur during cutting and (3) the thickness of the cutter is negligible. The cutting is considered to be applied along the y -axis at $x = x_c$. Assumption (1) introduces an unknown source of strength $Q_c(y, t)$ at x_c and together with assumption (2), the cutting problem can be described by the dimensionless 2D non-linear, unsteady, parabolic, heat conduction equation,

$$(1) \quad \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(k(u)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(k(u)\frac{\partial u}{\partial y}) + Q_c(y, t)\delta(x - x_c) \in D,$$

subject to initial condition $u(x, y, 0) = U_i(x, y)$, boundary conditions $u(0, y, t) = B_0(y, t)$, $u(1, y, t) = B_1(y, t)$, $u(x, 0, t) = C_0(x, t)$ and $u(x, 1, t) = C_1(x, t)$. Here $u(x, y, t)$ is the temperature distribution, $k(u)$ is the conductivity of the metal, $Q_c(y, t)$ is the unknown source being applied at $x = x_c$, $\delta(x - x_c)$ is the Dirac delta function and U_i , B_0 , B_1 , C_0 and C_1 are known functions.

Assumption (3) suggests the continuity of the function $\frac{\partial u}{\partial t}$ at $x = x_c$, which in turn suggests that $\int_{x_c^-}^{x_c^+} \frac{\partial u}{\partial t} dx = 0$. Here x_c^- denotes a spatial point just to the left of x_c and x_c^+ denotes a spatial point just to the right of x_c . Hence the equivalent source strength can be retrieved by integrating (1) from $x = x_c^-$ to $x = x_c^+$ to give

$$(2) \quad k(u)\frac{\partial u}{\partial x}|_{x_c^+} - k(u)\frac{\partial u}{\partial x}|_{x_c^-} + \frac{\partial}{\partial y}(k(u)\frac{\partial u}{\partial y})(x_c^+ - x_c^-) + Q_c(y, t) = 0$$

Assumption (3) also suggests that equation (2) can be truncated to:

$$(3) \quad k(u)\frac{\partial u}{\partial x}|_{x_c^+} - k(u)\frac{\partial u}{\partial x}|_{x_c^-} + Q_c(y, t) = 0$$

That is, heat fluxes just to the left and just to the right of x_c must be known. Temperature sensors are attached at $x = x_s$, such that $0 < x_s < x_c < 1$, and let the temperature measured by means of the temperature sensors be $u(x_s, y, t) = u^*(y, t)$. It is not necessary to have x_s being less than x_c . Similar problem partitioning can be generated for $0 < x_c < x_s < 1$. The measured temperatures are used to retrieve temperatures at the cutting points. Such inverse methods avoid the basic

difficulties of a direct method since remote temperatures can be measured more easily and accurately.

For computer simulation purposes, the sensor temperatures are modelled by the function $u^*(y, t) = \alpha y(y - 1)^2 \sin(\omega t)$. Its maximum value is generated by the amplitude, α . Its variation with respect to time is generated by the angular frequency ω .

3. Problem Partitioning

Problem partitioning is a DD method applied at the mathematical/physical problem level. In other words, decomposition is carried out by only considering the problem at this level [8]. In order to solve the inverse problem given in (1) with the additional condition available at $x = x_s$, problem partitioning is carried out to produce three sub-domains, such that each subproblem may define different numerical algorithms. The three sub-domains are, $D_1 = \{(x, y) : 0 < x < x_s \text{ and } 0 < y < 1\}$, $D_2 = \{(x, y) : x_s < x < x_c \text{ and } 0 < y < 1\}$, and $D_3 = \{(x, y) : x_c < x < 1 \text{ and } 0 < y < 1\}$. This problem partitioning removes the unknown source term $Q_c(y, t)$ and the Dirac delta function associated with it from the differential equations. The three sub-problems can be written as follows:

$$\begin{aligned} SP_1: \quad & \frac{\partial u_1}{\partial t} = \frac{\partial}{\partial x}(k(u_1)\frac{\partial u_1}{\partial x}) + \frac{\partial}{\partial y}(k(u_1)\frac{\partial u_1}{\partial y}) \in D_1 \\ & \text{subject to } u_1(x, y, 0) = U_i(x, y), u_1(0, y, t) = B_0(y, t), \\ & u_1(x_s, y, t) = u^*(y, t), u_1(x, 0, t) = C_0(x, t), u_1(x, 1, t) = C_1(x, t). \\ SP_2: \quad & \frac{\partial u_2}{\partial t} = \frac{\partial}{\partial x}(k(u_2)\frac{\partial u_2}{\partial x}) + \frac{\partial}{\partial y}(k(u_2)\frac{\partial u_2}{\partial y}) \in D_2 \\ & \text{subject to } u_2(x, y, 0) = U_i(x, y), u_2(x_s, y, t) = u^*(y, t), \\ & \frac{\partial u_2(x_s, y, t)}{\partial x} = \frac{\partial u_1(x_s, y, t)}{\partial x}, u_2(x, 0, t) = C_0(x, t), u_2(x, 1, t) = C_1(x, t). \\ SP_3: \quad & \frac{\partial u_3}{\partial t} = \frac{\partial}{\partial x}(k(u_3)\frac{\partial u_3}{\partial x}) + \frac{\partial}{\partial y}(k(u_3)\frac{\partial u_3}{\partial y}) \in D_3 \\ & \text{subject to } u_3(x, y, 0) = U_i(x, y), u_3(x_c, y, t) = u_2(x_c, y, t), \\ & u_3(1, y, t) = B_1(y, t), u_3(x, 0, t) = C_0(x, t), u_3(x, 1, y) = C_1(x, t). \end{aligned}$$

Since the temperature values are given at $y = 0$, $y = 1$, $x = 0$ and there are temperature sensors located at $x = x_s$, Dirichlet boundary conditions are defined at the boundary of D_1 . Solutions of the differential equation provide the required data to calculate the heat flux $\frac{\partial u}{\partial x}(x_s, y, t)$. Therefore, with the knowledge of the temperatures $u(x_s, y, t)$ acquired by the temperature sensors at $x = x_s$, an initial value problem can be formulated in D_2 . $u(x, y, t)$ values are obtained by solving this initial value problem. Finally, with the calculated temperatures $u(x_c, y, t)$, another Dirichlet problem can be formulated in D_3 . The above three subproblems are well-defined [1] [12], and a unique solution exists for each of them. The direct sum of these subproblem solutions gives the temperature distribution of the original problem, i.e.

$$(4) \quad u(x, y, t) = \begin{cases} u_1(x, y, t), & x \in D_1 \\ u_2(x, y, t), & x \in D_2 \\ u_3(x, y, t), & x \in D_3 \end{cases}.$$

3.1. Numerical schemes. To solve the problems in SP_1 and SP_3 a first order forward difference approximation of the temporal derivative and a second order FV approximation of the spatial derivatives are used. A five-point explicit scheme is produced from the approximation. Dropping the subscript used in denoting the sub-domains, the explicit scheme for the sub-domains D_1 and D_3 can be written as,

$$(5) \quad u_{i,j}^{(n+1)} = r_x b_i^{(n)} u_{i-1,j}^{(n)} + r_x a_i^{(n)} u_{i+1,j}^{(n)} + (1 - r_x a_i^{(n)} - r_x b_i^{(n)} - r_y c_j^{(n)} - r_y d_j^{(n)}) u_{i,j}^{(n)} + r_y d_j^{(n)} u_{i,j-1}^{(n)} + r_y c_j^{(n)} u_{i,j+1}^{(n)}$$

where (i, j) denotes the (i, j) -th grid point, $r_x = \frac{\Delta t}{(\Delta x)^2}$, $r_y = \frac{\Delta t}{(\Delta y)^2}$, $a_i^{(n)} = \frac{k_{i+1,j}^{(n)} + k_{i,j}^{(n)}}{2}$, $b_i^{(n)} = \frac{k_{i-1,j}^{(n)} + k_{i,j}^{(n)}}{2}$, $c_j^{(n)} = \frac{k_{i,j+1}^{(n)} + k_{i,j}^{(n)}}{2}$, $d_j^{(n)} = \frac{k_{i,j-1}^{(n)} + K_{i,j}^{(n)}}{2}$, (n) denotes the time-step, Δt is the step size along the temporal axis and Δx , Δy are the grid spacing along the spatial axis x , y , respectively. The initial value problem in SP_2 is solved by employing a second order Euler Predictor-Corrector (P-C) method along the x-axis for each time-step. Again, the spatial derivatives are discretised using second order FV approximations and the time derivative with a first order finite difference approximation. The two step P-C method can be written as:

$$(6) \quad \begin{pmatrix} u \\ v \end{pmatrix}^* = \begin{pmatrix} u \\ v \end{pmatrix} + \Delta x \underline{f}, \quad \begin{pmatrix} u \\ v \end{pmatrix}^{\text{new}} = \begin{pmatrix} u \\ v \end{pmatrix} + \frac{\Delta x}{2} \{ \underline{f} + \underline{f}^* \},$$

where $v = \frac{\partial u}{\partial x}$, $\underline{f} = \underline{f} \left(\begin{array}{c} u \\ v \end{array} \right) = \left(\begin{array}{c} v \\ \frac{1}{k(u)} \left(\frac{\partial u}{\partial t} - \frac{\partial}{\partial y} (k(u) \frac{\partial u}{\partial y}) - k'(u) v^2 \right) \end{array} \right)$ and $\underline{f}^* = \underline{f} \left(\begin{pmatrix} u \\ v \end{pmatrix}^* \right)$. A second order spatially accurate solution may be obtained for each of the three subproblems. Therefore, it is expected to have a second order spatially accurate global solution for the inverse problem (1). The effect of the local truncation error for SP_2 is minimised because of the small size of the sub-domain which usually consists of only a few Euler P-C steps. All experiments carried out gave stable results as long as the CFL condition $r_x, r_y \leq 0.25$ was satisfied.

3.2. Numerical results. Numerical results are obtained for equation (1) with $x_s = 0.5$, $x_c = 0.6$, $U_i(x, y) = 0$, $B_0(y, t) = 0$, $B_1(y, t) = 0$, $C_0(x, t) = 0$ and $C_1(x, t) = 0$. Sensor points are modelled as $u^*(y, t) = \alpha y(y-1)^2 \sin(\omega t)$, with $\alpha = 0.1$ and $\omega = 2\pi$. Non-linear heat conductivity is given by $k(u) = \frac{1}{1+u^2}$. Temperature distributions are shown for time $t = 0$ to $t = 0.5$ in Figure 1. The retrieved source/sink strength is also shown, Figure 2, it reflects the shape of the function used in the modelling of sensor temperatures, i.e. a sine function in time.

4. Exploiting Parallelism

Exploiting the parallel properties of an algorithm provides several key advantages. One of them is the expectation of a very fast execution of the algorithm. This in turn facilitates the real-time simulation (e.g. one-minute of temperature evolution is calculated using no more than one minute of computation time.). Another added advantage is that very large problem sizes (i.e. problem sizes that may not fit into the memory of a single-processing element) can be solved by using the total memory available from all the processors.

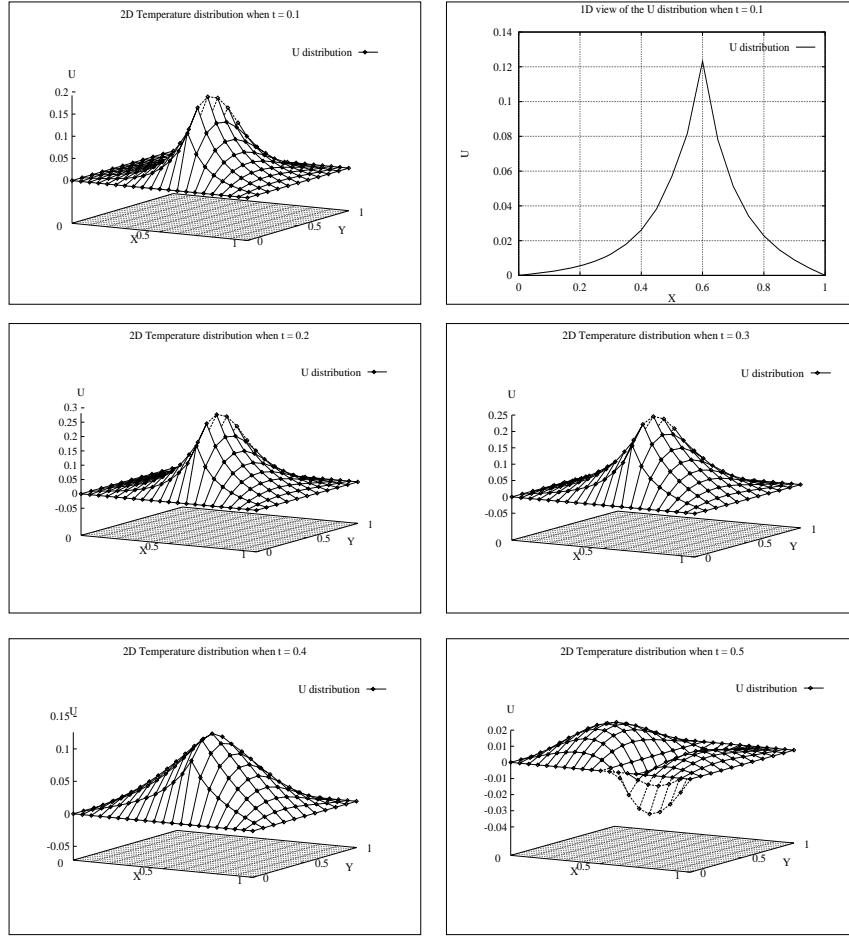


FIGURE 1. Temperature distributions from $t = 0.1$ to $t = 0.5$ and a 1D view at $t = 0.1$.

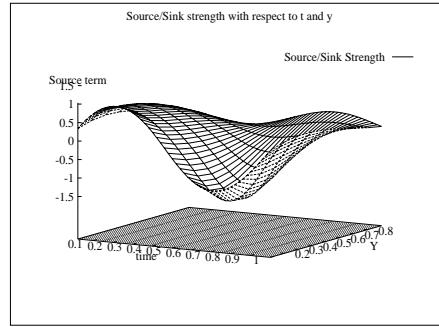


FIGURE 2. Source/Sink strength.

4.1. Domain-data parallelism. There are different ways of partitioning existing serial algorithms in order to utilise parallel and in particular distributed

environments. An advantage of using the DD approach to solve inverse problems is that the technique naturally provides a coarse-grained parallel algorithm. In other-words, each sub-domain generated due to the application of DD can be mapped directly to a processor and these sub-problems may be solved concurrently. In this paper this concept is referred to as “domain parallelism”. The calculation performed in sub-domain D_2 is much smaller (due to sensor and cutter points being very close to each other) than D_1 and D_3 . Also, calculations in D_1 can be carried out independently and gradients from D_2 and D_3 at $x = x_c$ are used to retrieve the source term. Considering these details, the calculations in sub-domains D_2 and D_3 are computed in one processor and the calculations in D_1 are computed in another. This minimises the communication and gives a better load balance among processors. That is, the domain parallelism requires two processors to solve the inverse problem defined by (1).

Domain parallelism has an obvious limitation in that it does not scale with an increasing number of processors. For the above cutting problem, only two processors are required. Data partitioning may be carried out within each sub-domain. In this paper we refer to this as “domain-data parallelism”. In partitioning the data the number of grid-points is divided amongst the processors as evenly as possible. If N denotes the total number of grid points and P denotes the number of processors and if $\frac{N}{P}$ is not an integer, then some processors will have more grid points than others. As a result, the remaining data is distributed as evenly as possible so as to reduce the load imbalance amongst the processors.

4.2. Parallel results. The domain-data parallel version of the numerical algorithm is implemented using FORTRAN 77 with MPI directives. The parallel implementation is tested using a loosely coupled and tightly coupled multi-processor environments. The loosely coupled environment used is, a set of Sun Sparc 5 workstations connected together by an Ethernet network. The SGI Origin 2000 machine describe the tightly coupled multi-processor environment. Performance of the parallel implementation on the two platforms is shown in Figure 3 and shows that, the trend is similar for both platforms. Differences in speedups between the platforms, appear significant for smaller problem sizes (e.g., for 10000 and 20000 mesh points). This is due to the differences in communication startup latencies and message transfer times between the platforms. The Origin 2000 has a very low startup latency and message transfer times relative to a network of Sun Sparc 5 stations. This difference becomes less significant with the larger problem sizes (e.g., for 80000 mesh points).

5. Conclusions

The use of a numerical algorithm developed by applying DD to the problem domain, in order to retrieve heat source/sink at the cutter and the calculation of the temperature distribution is presented. It is shown that good parallelism can be exploited from the DD based algorithm by using domain-data partitioning as the parallelisation strategy. MPI is used to investigate the parallel performance of the domain-data parallel version of the algorithm in a loosely coupled and tightly coupled multi-processor environments. The parallel performance results show that domain-data parallelism can be utilised effectively in both network clusters and tightly coupled multiprocessor machines.

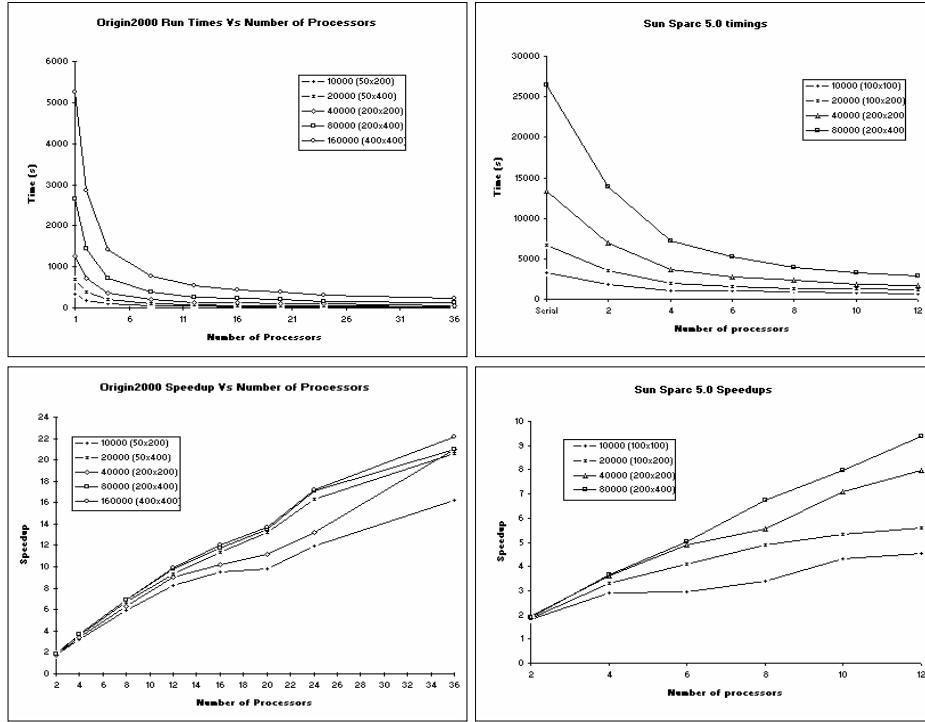


FIGURE 3. Parallel results for Origin 2000 and Sun Sparc 5s.

References

1. J. V. Beck, B. Blackwell, and C. R. St.Clair Jr, *Inverse heat conduction: Ill-posed problems*, Wiley-Interscience, 1985.
2. C-H.Lai, *A distributed algorithm for 1-d non-linear heat conduction with an unknown point source*, in proceedings of 9th International Conference on Domain Decomposition (To Appear).
3. N. D'Souza, *Numerical solution of one-dimensional inverse transient heat conduction by finite difference method*, A.S.M.E. (1975), Paper No. 68-WA-HT-81.
4. Message Passing Interface Forum, *Mpi: A message-passing interface standard*, June 1995, available from <http://www.mcs.anl.gov/mpi/>.
5. I. Frank, *An application of least squares method to the solution of the inverse problem of heat conduction*, Heat Transfer **85C** (1963), 378–379.
6. G. W. Krutz, R. J. Schoenhals, and P. S. Hore, *Application of finite element method to the inverse heat conduction problem*, Num. Heat Transfer **1** (1978), 489–498.
7. G. Krzysztof, M. C. Cialkowski, and H. Kaminski, *An inverse temperature field problem of the theory of thermal stresses*, Nucl. Eng. Des. **64** (1981), 169–184.
8. C-H. Lai, *Diakoptics, domain decomposition and parallel computing*, The Computer Journal **37** (1994), 840–846.
9. C-H. Lai and C. J. Palansuriya, *A distributed algorithm for the simulation of temperatures in metal cutting*, High Performance Computing and Networking, Lecture Notes in Computer Science **1067** (1996), 968–969.
10. G Stolz, Jr., *Numerical solutions to an inverse problem of heat conduction for simple shapes*, Heat Transfer **82** (1960), 20–26.
11. D. M. Trujillo, *Application of dynamic programming to the general inverse problem*, Int. j. numer. methods eng. **12** (1978), 613–624.
12. D. Zwillinger, *Handbook of differential equations*, Academic Press Inc., San Diego, 1989.

CENTRE FOR NUMERICAL MODELLING AND PROCESS ANALYSIS, UNIVERSITY OF GREENWICH,
LONDON SE18 6PF, UK
E-mail address: c.j.palansuriya@gre.ac.uk

CENTRE FOR NUMERICAL MODELLING AND PROCESS ANALYSIS, UNIVERSITY OF GREENWICH,
LONDON SE18 6PF, UK
E-mail address: c.h.lai@gre.ac.uk

CENTRE FOR NUMERICAL MODELLING AND PROCESS ANALYSIS, UNIVERSITY OF GREENWICH,
LONDON SE18 6PF, UK
E-mail address: c.s.ierotheou@gre.ac.uk

CENTRE FOR NUMERICAL MODELLING AND PROCESS ANALYSIS, UNIVERSITY OF GREENWICH,
LONDON SE18 6PF, UK
E-mail address: k.pericleous@gre.ac.uk

Overlapping Domain Decomposition and Multigrid Methods for Inverse Problems

Xue-Cheng Tai, Johnny Frøyen, Magne S. Espedal, and Tony F. Chan

1. Introduction

This work continues our earlier investigations [2], [3] and [8]. The intention is to develop efficient numerical solvers to recover the diffusion coefficient, using observations of the solution u , from the elliptic equation

$$(1) \quad -\nabla \cdot (q(x)\nabla u) = f(x) \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

Our emphasis is on the numerical treatment of discontinuous coefficient and efficiency of the numerical methods. It is well known that such an inverse problem is illposed. Its numerical solution often suffers from undesirable numerical oscillation and very slow convergence. When the coefficient is smooth, successful numerical methods have been developed in [5] [7]. When the coefficient has large jumps, the numerical problem is much more difficult and some techniques have been proposed in [2] and [3]. See also [9], [4] and [6] for some related numerical results in identifying some discontinuous coefficients.

The two fundamental tools we use in [2] and [3] are the total variation (TV) regularization technique and the augmented Lagrangian technique. The TV regularization allows the coefficient to have large jumps and at the same time it will discourage the oscillations that normally appear in the computations. The augmented Lagrangian method enforces the equation constraint in an H^{-1} norm and was studied in detail in [7]. Due to the bilinear structure of the equation constraint, the augmented Lagrangian reduces the output-least-squares (OLS) minimization to a system of coupled algebraic equations. How to solve these algebraic equations is of great importance in speeding up the solution procedure. The contribution of the present work is to propose an overlapping domain decomposition (DD) and a multigrid (MG) technique to evaluate the H^{-1} norm and at the same time use them as a preconditioner for one of the algebraic equations. Numerical tests will be given to show the speed-up using these techniques.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 35R30.
The fourth author was supported in part by NSF Grant ASC 9720257.

2. The augmented Lagrangian method

Let u_d be an observation for the solution u and \vec{u}_g be an observation for the gradient ∇u , both may contain random observation errors. Due to the illposedness of the inverse problem, it is often preferable to use the OLS minimization to recover $q(x)$. Let us define $K = \{q \mid q \in L^\infty(\Omega), 0 < k_1 \leq q(x) \leq k_2 < \infty\}$, with k_1 an k_2 known *a priori*, to be the admissible set for the coefficient. Let the mapping $e : K \times H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ be $e(q, u) = -\nabla \cdot (\nabla q \nabla u) - f$, which is the equation constraint. We shall use $R(q) = \int_{\Omega} \sqrt{|\nabla q|^2 + \epsilon} dx$, which approximate the TV-norm of $q(x)$, as the regularization term. In our experiments, the value of ϵ is always taken in $\epsilon \in [0.001, 0.01]$. The OLS minimization can be written

$$(2) \quad \min_{e(q,u)=0, q \in K} \frac{1}{2} \|u - u_d\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\nabla u - \vec{u}_g\|_{L^2(\Omega)}^2 + \beta R(q).$$

For more details on numerical approximations of the TV-norms, we refer to Chan and Tai [3]. As the inverse problem is illposed, its numerical solution is very sensitive to the observation errors. When the observation errors are very large, we must use proper noise removal procedure, see section 4 of [3] for the detailed algorithms that remove noise from the observations.

The Lagrangian method is often used for minimization problems with equality constraint. However, the augmented Lagrangian method is better when the minimization problem is illposed or the Hessian matrix of the cost functional has very small positive eigenvalues. For minimization (2), the associated augmented Lagrangian functional is

$$(3) \quad \begin{aligned} L_r(q, u, \lambda) = & \frac{1}{2} \|u - u_d\|_{L^2(\Omega)}^2 + \frac{1}{2} \|\nabla u - \vec{u}_g\|_{L^2(\Omega)}^2 + \beta R(q) \\ & + \frac{r}{2} \|e(q, u)\|_{H^{-1}(\Omega)}^2 + (\lambda, e(q, u))_{H^{-1}(\Omega)}, \\ & \forall q \in K, u \in H_0^1(\Omega), \lambda \in H^{-1}(\Omega). \end{aligned}$$

The following algorithm is used to find a saddle point for $L_r(q, u, \lambda)$:

Algorithm

Step 1 Choose $u_0 \in H_0^1(\Omega)$, $\lambda_0 \in H^{-1}(\Omega)$ and $r > 0$.

Step 2 Set $u_n^0 = u_{n-1}$. For $k = 1, 2, \dots, k_{max}$, do:

Step 2.1 Find $q_n^k \in K$ such that

$$(4) \quad q_n^k = \arg \min_{q \in K} L_r(q, u_n^{k-1}, \lambda_{n-1}).$$

Step 2.2 Find $u_n^k \in H_0^1(\Omega)$ such that

$$(5) \quad u_n^k = \arg \min_{u \in H_0^1(\Omega)} L_r(q_n^k, u, \lambda_{n-1}).$$

Step 3 Set $u_n = u_n^k$, $q_n = q_n^k$, and update λ_n as $\lambda_n = \lambda_{n-1} + r e(q_n, u_n)$.

In our simulations, we take $k_{max} = 2$. The above algorithm has a linear rate of convergence, see [7]. Second order scheme can also be used to search for a saddle point, see [7, p.98]. If (q^*, u^*, λ^*) is a saddle point of L_r , then (q^*, u^*) is a minimizer of (2). For fixed u_n^{k-1} and q_n^k , the minimization problems of Step 2.1 and Step 2.2 are equivalent to two algebraic equations. See [3] for the detailed matrix representation of the corresponding algebraic equations. Problems (4) and (5) are solved by direct solver in [2] and [3]. The numerical accuracy and the executing time is superior to earlier literature results. However, we must improve the efficiency and use some

iterative solvers in order to be able to solve real life large size problems. Without using iterative solvers, the memory limit will prevent us from doing simulations for real life three dimensional problems. To use an iterative solver, the rate of convergence of the iterative solver is of great concern for the efficiency of the whole algorithm. Moreover, the way that the H^{-1} -norm is evaluated is very critical in avoiding the solving of large size sparse matrices in the iterative procedure. Let Δ denote the Laplace operator, which is a homeomorphism from $H_0^1(\Omega)$ to $H^{-1}(\Omega)$. It is true that $\|\Delta^{-1}f\|_{H_0^1(\Omega)} = \|f\|_{H^{-1}(\Omega)}$. Thus, we need to invert a sparse matrix Δ in order to compute the H^{-1} -norm. However, we can obtain the H^{-1} -norm of f by inverting smaller size matrices with the domain decomposition methods or using multigrid type methods to avoid inverting any matrices by using the theorem of the next section.

3. Space decomposition methods

Recent research reveals that both domain decomposition and multigrid type methods can be analysed using the frame work of space decomposition and subspace correction, see Chan and Sharapov [1], Tai and Espedal [11] [12], Tai [10] and Xu [14]. In this section, we show that we can use them to evaluate the H^{-1} -norm.

We present the results for a general Hilbert space and for general space decomposition techniques. For a given Hilbert space V , we denote V^* as its dual space and use (\cdot, \cdot) to denote its inner product. Notation $\langle \cdot, \cdot \rangle$ is used to denote the duality pairing between V and V^* . We consider the case where V can be decomposed as a sum of subspaces:

$$V = V_1 + V_2 + \cdots + V_m.$$

Moreover, we assume that there is a constant $C_1 > 0$ such that $\forall v \in V$, we can find $v_i \in V_i$ that satisfy:

$$(6) \quad v = \sum_{i=1}^m v_i, \quad \text{and} \quad \left(\sum_{i=1}^m \|v_i\|_V^2 \right)^{\frac{1}{2}} \leq C_1 \|v\|_V$$

and there is an $C_2 > 0$ such that

$$(7) \quad \sum_{i=1}^m \sum_{j=1}^m (v_i, v_j) \leq C_2 \left(\sum_{i=1}^m \|v_i\|_V^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^m \|v_i\|_V^2 \right)^{\frac{1}{2}}, \quad \forall v_i \in V_i \quad \text{and} \quad \forall v_j \in V_j.$$

THEOREM 1. *Assume the decomposed spaces satisfy (6) and (7), then*

$$(8) \quad \frac{\|f\|_{V^*}}{C_1} \leq \left(\sum_{i=1}^m \|f\|_{V_i^*}^2 \right)^{\frac{1}{2}} \leq C_2 \|f\|_{V^*} \quad \forall f \in V^* \subset V_i^*.$$

Details of the proof of the above theorem will be given in a forthcoming paper. The theorem shows that in order to get $\|f\|_{V^*}$, we just need to use some parallel processors to compute $\|f\|_{V_i^*}^2$ and sum them together. For domain decomposition methods, we need to invert some smaller size matrices to get $\|f\|_{V_i^*}^2$. For multigrid methods, no matrices need to be inverted.

4. Numerical Tests

Let $\Omega = [0, 1] \times [0, 1]$. For a given f and piecewise smooth q , we compute the true solution from (1) and get its gradient ∇u . Let R_d and \vec{R}_g be vectors of random

numbers between $[-1/2, 1/2]$. The observations are generated by

$$(9) \quad u_d = u + \delta R_d \|u_d\|_{L^2(\Omega)}, \quad \vec{u}_g = \nabla u + \delta \vec{R}_g \|\vec{u}_g\|_{\mathbf{L}^2(\Omega)}.$$

We shall use finite element (FE) approximations. The domain Ω is first divided into subdomains $\Omega_i, i = 1, 2, \dots, m$ with diameters of the size H , which will also be used as the coarse mesh elements. Each subdomain is refined to form a fine mesh division for Ω of mesh parameter h ($h \ll H$). Each subdomain Ω_i is extended by a size $\delta = cH$ ($0 < c < 1$) to get overlapping subdomains Ω_i^δ . Let $S_0^h(\Omega)$, $S_0^h(\Omega_i^\delta)$ and $S_0^H(\Omega)$ be the bilinear FE spaces with zero traces on the corresponding boundaries on the fine mesh, subdomain Ω_i^δ and coarse mesh respectively. It is true that

$$S_0^h(\Omega) = S_0^H(\Omega) + \sum_{i=1}^m S_0^h(\Omega_i^\delta).$$

For the above decomposition, the constants C_1 and C_2 do not depend on the mesh parameters h and H , see [14]. Estimate (8) shows that we only need to invert the matrices associated with the subdomains and the coarse mesh to get the H^{-1} norms.

In order to use multigrid type techniques, we take Ω as the coarsest mesh and use rectangular elements. At a given level, we refine each element into four elements by connecting the midpoints of the edges of the rectangles of a coarser grid. Starting from Ω and repeating the above procedure J times, we will get J levels of meshes. Let $V_k, k = 1, 2, \dots, J$ be the bilinear FE spaces over the levels and denote $\{\phi_i^k\}_{i=1}^{n_k}$ the interior nodal bases for the k^{th} level FE space, it is easy to see that

$$V = \sum_{k=1}^J \sum_{i=1}^{n_k} V_i^k \quad \text{with} \quad V = V_J, V_i^k = \text{span}(\phi_i^k).$$

For the multigrid decomposition, the subspaces V_i^k are one dimensional and the constants C_1, C_2 are independent of the mesh parameters and the number of levels. No matrix need to be inverted to get the H^{-1} norms.

The bilinear FE spaces introduced above will be used as the approximation spaces for u and λ . Piecewise constant FE functions on the fine mesh are used to approximate the coefficient q . For a given q, u and λ , let $B(q, u, \lambda)$ and $A(q, \lambda)$ be the matrices that satisfy

$$\partial L_r(q, u, \lambda) / \partial q = B(q, u, \lambda)q, \quad \partial L_r(q, u, \lambda) / \partial u = A(q, \lambda)u.$$

Let $B_n^k = B(q_n^k, u_n^{k-1}, \lambda_{n-1})$ and $A_n^k = A(q_n^k, \lambda_{n-1})$. Assume that the solution of (4) is in the interior of K , then (4) and (5) are equivalent to solving

$$(10) \quad a) \quad B_n^k q_n^k = \alpha_n^k, \quad b) \quad A_n^k u_n^k = \beta_n^k.$$

with some known vector α_n^k and β_n^k . Due to the regularization term $R(q)$, the matrix B_n^k depends on q_n^k . A simple linearization procedure is employed to deal with the nonlinearity, see [3, 13]. If we use conjugate gradient (CG) method to solve the equations (10), it is not necessary to know the matrices, we just need to calculate the product of the matrices with given vectors. It is easy to see that the equations in (10) have symmetric and positive define matrices. We use CG without preconditioner to solve (10.a) and use a preconditioned CG to solve (10.b). The stopping criteria for the CG iterations is that the residual has been reduced by a factor of 10^{-10} or the iteration number has reached 300 (In Table 2 the maximum iteration number is 5000 in order to see the CPU time usage for bad β). The

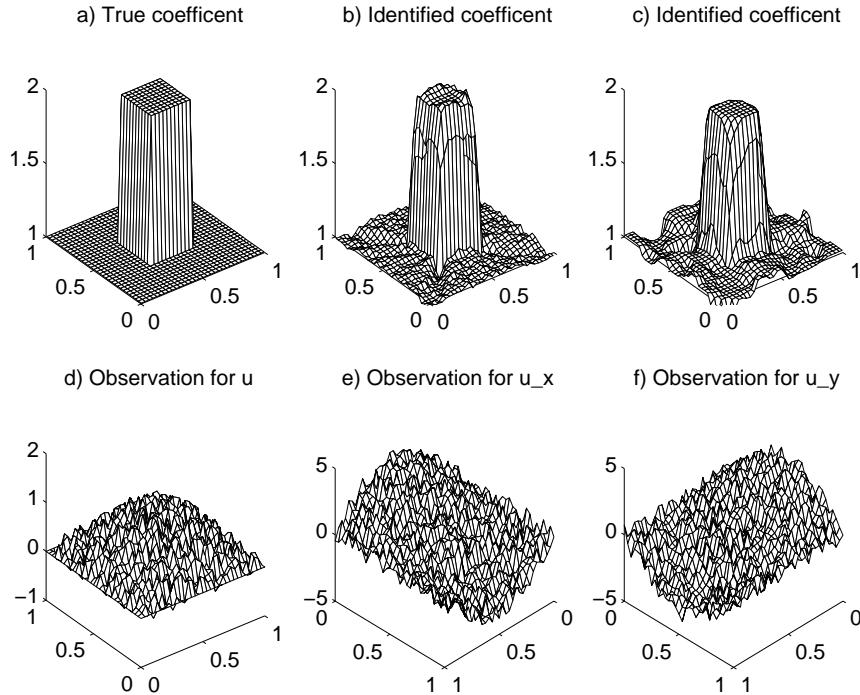


FIGURE 1. The identified coefficients and the observation data.

constants k_1 and k_2 used for defining the admissible coefficient set are taken to be $k_1 = 0$, $k_2 = \infty$. The simulations are tested with a sequential machine in programming language C++. Three different methods are used for the preconditioner and for evaluating the H^{-1} norms:

1. Do LU decomposition (LU-D) for the Laplace operator and use it as the preconditioner and also for the H^{-1} norms.
2. Use domain decomposition for the preconditioning and also for the H^{-1} norms.
3. Use multigrid for the preconditioning and also for the H^{-1} norms.

As the tests are done with a sequential machine, the multiplicative version of the MG and DD methods are used. In Table 1, the CPU time for different iteration numbers is given. We have used $\beta = 0.00125$ and $h = 1/128$. Here and later h and nx denote the mesh size and number of elements used both for the x - and y -directions, respectively. We observe that the domain decomposition approach and the multigrid approach are much faster than the LU-decomposition. The multigrid method is slightly better than the domain decomposition method. The identified coefficients and the observations are shown in Figure 1. In identifying the coefficient in subfigure 1.b), we have added 10% of noise and used $h = 1/128$, $\beta = 0.00025$, $r = 100$. At iteration 20, $\|q_n - q\|_{L^2(\Omega)} = 0.0631$ and $\|e(q_n, u_n)\|_{L^2(\Omega)} = 1.5 \times 10^{-7}$. In subfigure 1.c), the identified coefficient is with noise level $\delta = 100\%$ and we have used $h = 1/128$, $\beta = 0.03125$, $r = 100$. At iteration 20, $\|q_n - q\|_{L^2(\Omega)} = 0.1013$ and $\|e(q_n, u_n)\|_{L^2(\Omega)} = 1.1 \times 10^{-6}$.

TABLE 1. CPU time (in sec.) versus iteration with $\beta = 0.00125$, $h = 1/128$.

<i>iter</i> =	MG	DD	LU-D
1	31.87	94.04	942.76
3	121.27	370.15	3350.17
5	221.68	668.06	5999.82
7	327.14	954.52	8749.58
9	433.74	1244.67	11509.80
11	541.47	1538.97	14227.90
12	653.80	1836.38	16935.80
15	777.22	2137.01	19628.30
17	885.29	2439.20	22309.40
20	1043.60	2887.72	26300.10

TABLE 2. CPU time (in Sec.) versus β with $h = 1/64$, *iteration* = 20.

β =	MG	DD	LU-D
0.00001	1309.67	3270.60	5009.87
0.00005	544.13	1320.36	2146.69
0.00025	256.89	758.76	1107.20
0.00125	173.72	431.31	747.76
0.00625	172.15	459.70	728.08
0.03125	224.22	566.91	922.79

TABLE 3. CPU time (in Sec.) versus $h = 1/nx$ with $\beta = 0.00125$, *iteration* = 20.

nx =	MG	DD	LU-D
16	6.06	16.83	9.88
32	24.56	60.89	61.71
64	171.76	406.91	720.45
128	1393.95	3859.8	35258.20

The regularization parameter β is introduced to prevent numerical oscillations. If it is chosen to be big, the discontinuity is smeared out and large errors are introduced. If it is chosen to be too small, it can not control the numerical oscillations and so prevent us from getting accurate numerical solutions. From our numerical tests, we find that the value of β is also of critical importance for the rate of convergence for the CG method. In Table 2, the CPU time in seconds for different values of β is compared for the three different approaches. It is clear that very small or very large β increases the computing time.

Table 3 is used to show the CPU time usage for different mesh sizes h . Let us note that the finest mesh is of size $h = 1/128$ with a total number of grid points $128 \times 128 \approx 2 \times 10^4$. For inverse problems we considered here, there are not many numerical approaches that can handle such a large number of unknowns. It shall

also be noted that 100% of observation errors are added to the observations, i.e. $\delta = 100\%$ in (9), see d) e) f) of Figure 1.

References

1. T. F. Chan and I. Sharapov, *Subspace correction multilevel methods for elliptic eigenvalue problems*, Proceedings of the 9th international domain decomposition methods (P. Bjørstad, M. Espedal, and D. Keyes, eds.), John Wiley and Sons, To appear.
2. T. F. Chan and X.-C. Tai, *Augmented Lagrangian and total variation methods for recovering discontinuous coefficients from elliptic equations*, Computational Science for the 21st Century (M. Bristeau, G. Etgen, W. Fitzgibbon, J. L. Lions, J. Periaux, and M. F. Wheeler, eds.), John Wiley & Sons, 1997, pp. 597–607.
3. ———, *Identification of discontinuous coefficient from elliptic problems using total variation regularization*, Tech. Report CAM-97-35, University of California at Los Angeles, Department of Mathematics, 1997.
4. Z. Chen and J. Zou, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, Tech. Report Report 97-06, The Chinese University of Hong Kong, 1997.
5. K. Ito and K. Kunisch, *The augmented Lagrangian method for parameter estimation in elliptic systems*, SIAM J. Control Optim. **28** (1990), 113–136.
6. Y. L. Keung and J. Zou, *Numerical identifications of parameters in parabolic systems*, Tech. Report Report 97-16, The Chinese University of Hong Kong, 1997.
7. K. Kunisch and X.-C. Tai, *Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces*, SIAM J. Numer. Anal. **34** (1997), 91–118.
8. ———, *Nonoverlapping domain decomposition methods for inverse problems*, Proceedings of the 9th international domain decomposition methods (P. Bjørstad, M. Espedal, and D. Keyes, eds.), John Wiley and Sons, To appear.
9. T. Lin and E. Ramirez, *A numerical method for parameter identification of a two point boundary value problem*, 1997.
10. X.-C. Tai, *Parallel function and space decomposition methods*, The Finite element Method: Fifty years of the Courant Element (P. Neittaanmaki, ed.), Lecture notes in pure and applied mathematics, vol. 164, Marcel Dekker, 1994, pp. 421–432.
11. X.-C. Tai and M. Espedal, *Rate of convergence of a space decomposition method for linear and nonlinear problems*, SIAM J. Numer. Anal. (1998), to appear.
12. ———, *A space decomposition method for minimization problems*, Proceedings of the 9th international domain decomposition methods (P. Bjørstad, M. Espedal, and D. Keyes, eds.), John Wiley and Sons, To appear.
13. C. Vogel and M. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comp. **17** (1996), 227–238.
14. J. C. Xu, *Iteration methods by space decomposition and subspace correction*, SIAM Rev. **34** (1992), 581–613.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BERGEN, JOHANNES BRUNSGATE 12, 5008, BERGEN, NORWAY.

E-mail address: Xue-Cheng.Tai@mi.uib.no

URL: <http://www.mi.uib.no/~tai>.

RF-ROGALAND RESEARCH, THORMØLLENSGT. 55, 5008 BERGEN, NORWAY.

E-mail address: Johnny.Froyen@rf.no.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BERGEN, JOHANNES BRUNSGATE 12, 5008, BERGEN, NORWAY.

E-mail address: Magne.Espedal@mi.uib.no

URL: <http://www.mi.uib.no/~resme>.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, 405 HILGARD AVENUE, LOS ANGELES, CA 90095-1555.

E-mail address: chan@math.ucla.edu

URL: <http://www.math.ucla.edu/~chan>.

Some Results on Schwarz Methods for a Low-Frequency Approximation of Time-Dependent Maxwell's Equations in Conductive Media

Andrea Toselli

1. Introduction

In this paper, we recall some recent theoretical results on two-level overlapping Schwarz methods for finite element approximations of Maxwell's equations and present some numerical results to compare the performances of different overlapping Schwarz methods, when varying the number of subregions, the mesh size of the coarse space and the time step of the implicit finite difference scheme employed.

When studying low-frequency electromagnetic fields in conductive media, the displacement current term in Maxwell's equations is generally neglected and a parabolic partial differential equation is solved. The electric field \mathbf{u} satisfies the equation

$$(1) \quad \operatorname{curl}(\mu^{-1} \operatorname{curl} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = -\frac{\partial \mathbf{J}}{\partial t}, \text{ in } \Omega,$$

where $\mathbf{J}(\mathbf{x}, t)$ is the current density and μ and σ are the magnetic permeability and the electric conductivity of the medium. For their meaning and for a general discussion of Maxwell's equations, see [3]. Here Ω is a bounded, three-dimensional polyhedron, with boundary Γ and outside normal \mathbf{n} . For a perfect conducting boundary, the electric field satisfies the essential boundary condition

$$(2) \quad \mathbf{u} \times \mathbf{n}|_{\Gamma} = 0.$$

For the analysis and solution of Maxwell's equations suitable Sobolev spaces must be introduced. The space $H(\operatorname{curl}, \Omega)$, of square integrable vectors, with square integrable curls, is a Hilbert space with the scalar product

$$(3) \quad a(\mathbf{u}, \mathbf{v}) = (\operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v}) + (\mathbf{u}, \mathbf{v}),$$

where (\cdot, \cdot) denotes the scalar product in $L^2(\Omega)$. The subspace of $H(\operatorname{curl}, \Omega)$ of vectors with vanishing tangential component on Γ is denoted by $H_0(\operatorname{curl}, \Omega)$. For the properties of $H(\operatorname{curl}, \Omega)$ and $H_0(\operatorname{curl}, \Omega)$, see [5].

In the following section, we recall the variational problem associated with (1) and (2), and the finite element spaces employed for its approximation. In Section 3,

1991 *Mathematics Subject Classification*. Primary 65M60; Secondary 65C20, 35Q60, 65M55.

The author was supported in part by the National Science Foundation under Grant NSF-ECS-9527169 and in part by the U.S. Department of Energy under Contract DE-FG02-92ER25127.

we describe the two-level Schwarz method studied and state some theoretical results about its convergence properties. Section 4 is devoted to the numerical results.

2. Discrete problem

When an implicit FD scheme is employed and a finite element space $V \subset H_0(\text{curl}, \Omega)$ is introduced, equations (1) and (3) can be approximated by:

Find $\mathbf{u} \in V$ such that

$$(4) \quad a_\eta(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}) + \eta (\mathbf{curl} \mathbf{u}, \mathbf{curl} \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in V,$$

at each time step; η is a positive quantity, proportional to the time step Δt , and f depends on the solution at the previous steps, as well as on the right hand side of (1). See [1] for the finite element approximation of Maxwell's equations, and [10] for the time approximation of parabolic problems.

For the finite element approximation, we first consider a shape-regular triangulation \mathcal{T}_h of the domain Ω , consisting of tetrahedra. Here h is the maximum diameter of \mathcal{T}_h . We employ the Nédélec spaces of the first kind, of degree $k > 0$, which were introduced in [8]; see also [5]. Other choices of finite element spaces are also possible, see [9], as well as triangulations made of hexahedra and prisms, see [8], [9].

Let $V \subset H_0(\text{curl}, \Omega)$ be the Nédélec space of degree k , built on \mathcal{T}_h , of vectors with vanishing tangential component on the boundary. We recall that vectors in V are not continuous, in general, but that only the continuity of their tangential component is preserved across the faces of the tetrahedra, as it is physically required for the electric field. The degrees of freedom associated to this finite element space involve integrals of the tangential components over the edges and the faces, as well as moments computed over each tetrahedron. See [8], [9], [5] and the references in [12] and [7] for further details on Nédélec spaces.

3. Additive two-level method

In order to build a two-level overlapping preconditioner for (4), we have to introduce an additional coarse triangulation and a decomposition of Ω into overlapping subregions. This can be done in a standard way; see [11] as a general reference for this section.

We suppose that the triangulation \mathcal{T}_h is obtained by refining a shape-regular coarse triangulation \mathcal{T}_H , $H > h$, made of tetrahedra $\{\Omega_i\}_{i=1}^J$. Let us consider a covering of Ω , say $\{\Omega'_i\}_{i=1}^J$, such that each open set Ω'_i is the union of tetrahedra of \mathcal{T}_h and contains Ω_i . Define the overlapping parameter δ by

$$\delta = \min_i \{\text{dist}(\partial\Omega'_i, \Omega_i)\}.$$

We suppose that $\delta \geq \alpha H$, for a constant $\alpha > 0$ (generous overlap), and that every point $P \in \Omega$ belongs to at most N_c subregions (finite covering).

The coarse space V_0 is defined as the Nédélec space over the coarse triangulation \mathcal{T}_H , $H > h$, and the local spaces $V_i \subset V$, associated to the subregions $\{\Omega'_i\}_{i=1}^J$, are obtained by setting the degrees of freedom outside Ω'_i to zero. Since the triangulations \mathcal{T}_h and \mathcal{T}_H are nested, the coarse space V_H is contained in V ; see [6]. The space V admits the decomposition $V = \sum_{i=0}^J V_i$.

Let us now define the following projections for $i = 0, \dots, J$:

$$(5) \quad T_i : V \longrightarrow V_i,$$

$$(6) \quad a_\eta(T_i \mathbf{u}, \mathbf{v}) = a_\eta(\mathbf{u}, \mathbf{v}), \forall \mathbf{v} \in V_i,$$

where $a_\eta(\cdot, \cdot)$ is defined in (4) and let us introduce the additive Schwarz operator

$$(7) \quad T = \sum_{j=0}^J T_j : V \longrightarrow V.$$

We solve the equation

$$T \mathbf{u} = \mathbf{g},$$

with the conjugate gradient method, without any further preconditioner, employing $a_\eta(\cdot, \cdot)$ as the inner product and a suitable right hand side \mathbf{g} ; see [4], [11].

Two-level multiplicative schemes can also be designed; see [4], [11]. The error \mathbf{e}_n at the n -th step satisfies the equation

$$(8) \quad \mathbf{e}_{n+1} = E \mathbf{e}_n = (I - T_J) \cdots (I - T_0) \mathbf{e}_n, \forall n \geq 0.$$

Different choices of multiplicative and hybrid operators are also possible and Krylov subspace methods, such as GMRES, can be employed as accelerators; see [11] for a more detailed discussion. A hybrid method will be considered in the next section.

The main result concerning the convergence properties of the standard additive and multiplicative algorithms is contained in the following theorem:

THEOREM 1. *If the domain Ω is convex, and the triangulations \mathcal{T}_h and \mathcal{T}_H shape-regular and quasiuniform, then the condition number of the additive algorithm and the norm of the error operator E are bounded uniformly with respect to h , the number of subregions and η . The bounds increase with the inverse of the relative overlap H/δ and the finite covering parameter N_c .*

A proof of this theorem can be found in [12], for the case $\eta = 1$, and in [7] for the general case. We remark that the proofs employ the discrete Helmholtz decomposition of Nédélec spaces and suitable projections onto the spaces of discrete divergence-free functions; see [5] and [6]. The convexity of the domain Ω and the quasiuniformity of the meshes appear to be necessary for the error bounds of discrete divergence-free vectors.

4. Numerical results

In this section we present some numerical results to analyze the dependence of some Schwarz algorithms on the overlap, the number of subdomains and the time step. We will also show some results for a non-convex domain, for which the analysis carried out in [12] and [7] is not valid. We consider the Dirichlet problem (4) and use hexahedral Nédélec elements.

We have tested the following Schwarz algorithms:

- (i) The Conjugate gradient method applied to the *additive one-level* operator

$$T_{as1} = \sum_{i=1}^J T_i,$$

where the T_i are the projections onto the local subspaces, defined in (5), (6).

TABLE 1. Estimated condition number and N_c (in parenthesis), versus H/δ and the number of subregions: additive one-level algorithm, $\Omega = (0, 1)^3$, $\eta = 1$, $16 \times 16 \times 16$ -element fine mesh (13,872 unknowns).

	$2 \times 2 \times 2$	$4 \times 4 \times 4$	$8 \times 8 \times 8$
$H/\delta = 8$	28.7 (8)	-	-
$H/\delta = 4$	14.3 (8)	40.0 (8)	-
$H/\delta = 8/3$	10.1 (8)	-	-
$H/\delta = 2$	8.62 (8)	13.5 (8)	46.4 (8)
$H/\delta = 4/3$	8.02 (8)	27.0 (27)	-
$H/\delta = 1$	-	5.19 (27)	15.5 (27)

TABLE 2. Estimated condition number and N_c (in parenthesis), versus H/δ and the number of subregions: additive two-level algorithm, $\Omega = (0, 1)^3$, $\eta = 1$, $16 \times 16 \times 16$ -element fine mesh (13,872 unknowns).

	$2 \times 2 \times 2$	$4 \times 4 \times 4$	$8 \times 8 \times 8$
$H/\delta = 8$	15.7 (8)	-	-
$H/\delta = 4$	9.67 (8)	10.3 (8)	-
$H/\delta = 8/3$	8.48 (8)	-	-
$H/\delta = 2$	8.42 (8)	8.91 (8)	9.23 (8)
$H/\delta = 4/3$	8.73 (8)	27.0 (27)	-
$H/\delta = 1$	-	18.2 (27)	26.3 (27)

(ii) The Conjugate gradient method applied to the *additive two-level* operator

$$T_{as2} = T_0 + \sum_{i=1}^J T_i,$$

where the T_0 the projection onto the coarse space V_0 .

(iii) The GMRES method applied to the non-symmetric *two-level hybrid* operator

$$T_{hy} = I - (I - T_0) \left(I - \omega \sum_{i=1}^J T_i \right),$$

where ω is a scaling parameter.

Our results have been obtained on a SUN Ultra1, using the PETSc 2.0 library; see [2].

The independence of the condition number on the diameter h of the fine mesh is observed, when the number of subdomains and the relative overlap δ/H are fixed; the results are not presented here.

We first consider a unit cube Ω , with a fixed value of $\eta = 1$. Tables 1 and 2 show the estimated condition numbers for algorithms (i) and (ii), as functions of the relative overlap and the number of subregions; in parenthesis, we also show the value of N_c defined in the previous section. We observe:

TABLE 3. Number of iterations versus H/δ and the number of subregions, to reduce the residual error by a factor 10^{-6} : additive two-level algorithm (ii), $\Omega = (0, 1)^3$, $\eta = 1$, $16 \times 16 \times 16$ -element fine mesh (13,872 unknowns).

	8	64
$H/\delta = 8$	26	-
$H/\delta = 4$	24	22
$H/\delta = 8/3$	23	-
$H/\delta = 2$	21	21
$H/\delta = 4/3$	21	33
$H/\delta = 1$	-	27

TABLE 4. Number of iterations and residual error (in parenthesis), versus H/δ and the number of subregions, to reduce the residual error of the preconditioned system by a factor 10^{-9} : hybrid two-level algorithm (iii) with optimal scaling, $\Omega = (0, 1)^3$, $\eta = 1$, $16 \times 16 \times 16$ -element fine mesh (13,872 unknowns).

	8	64
$H/\delta = 8$	$28 (4.5 \times 10^{-3})$	-
$H/\delta = 4$	$26 (1.6 \times 10^{-3})$	$26 (9.8 \times 10^{-5})$
$H/\delta = 8/3$	$24 (7.3 \times 10^{-5})$	-
$H/\delta = 2$	$23 (1.0 \times 10^{-5})$	$24 (8.6 \times 10^{-5})$
$H/\delta = 4/3$	$20 (2.8 \times 10^{-5})$	$50 (1.3 \times 10^{-7})$
$H/\delta = 1$	-	$20 (4.8 \times 10^{-6})$

- For both algorithms, the condition number decreases with H/δ decreasing, when the number of subregions and N_c are fixed. In accordance with the analysis in [7], the condition number increases with N_c ; the bound of the largest eigenvalue of the Schwarz operators grows linearly with N_c .
- For a fixed value of the relative overlap and N_c , the condition number grows rapidly with the number of subregions for the one-level algorithm, while it grows slowly for the two-level case.
- The two-level algorithm behaves better than the one-level method when the overlap is small, but seems more sensitive to N_c .

For the same domain $\Omega = (0, 1)^3$ and the same value of $\eta = 1$, Tables 3 and 4 show the number iterations for algorithms (ii) and (iii), as functions of H/δ , for the cases of 8 and 64 subregions. In order to compare the two methods, a reduction by a factor 10^{-6} of the residual error of the unpreconditioned system was chosen for the CG algorithm in (ii), while a reduction by a factor 10^{-9} of the residual error of the preconditioned system was considered in (iii). In Table 4, we also show the residual error of the unpreconditioned system in parenthesis. GMRES was restarted each 30 iterations.

The results show that algorithm (iii) gives a number of iterations that is comparable to the ones of (ii). According to our numerical tests, the optimal value of the scaling factor ω depends on the overlap and the number of subregions. The results

TABLE 5. Estimated condition number versus H/δ and the number of subregions: additive one-level algorithm, $\Omega = (0, 1)^3 \setminus [0, 1/2]^3$, $16 \times 16 \times 16$ -element fine mesh (12, 336 unknowns).

	$2 \times 2 \times 2$	$4 \times 4 \times 4$	$8 \times 8 \times 8$
$H/\delta = 8$	31.5	-	-
$H/\delta = 4$	15.0	41.2	-
$H/\delta = 2$	8.65	13.6	48.9
$H/\delta = 1$	-	5.84	18.5

TABLE 6. Estimated condition number versus H/δ and the number of subregions: additive two-level algorithm, $\Omega = (0, 1)^3 \setminus [0, 1/2]^3$, $\eta = 1$, $16 \times 16 \times 16$ -element fine mesh (12, 336 unknowns).

	$2 \times 2 \times 2$	$4 \times 4 \times 4$	$8 \times 8 \times 8$
$H/\delta = 8$	52.2	-	-
$H/\delta = 4$	43.4	17.3	-
$H/\delta = 2$	36.2	17.1	10.2
$H/\delta = 1$	-	24.8	27.2

in Table 4 were obtained with optimal scaling. We also ran some tests with a symmetrized version of the hybrid algorithm, obtained by adding another smoothing step on the subdomains, see [11], but the results are not so good as in (iii) and are not presented here. The full multiplicative preconditioner was not implemented in the PETSc version that we used.

We have also considered the case $\Omega = (0, 1)^3 \setminus [0, 1/2]^3$, in which Ω is not convex and for which the analysis carried out in [12] and [7] is not valid. Tables 5 and 6 show the estimated condition numbers for algorithms (i) and (ii), as functions of the relative overlap and the number of subregions; in order to facilitate the comparison, we have shown the number of subregions and elements corresponding to the discretization of the whole $(0, 1)^3$. We observe:

- All the remarks made for the case $\Omega = (0, 1)^3$ are still valid, in general, except that the one-level algorithm behaves better than the two-level one, unless the number of subregions is large.
- The one-level algorithm shows a slight increase of the condition number, compared to the convex case, while a considerable deterioration of the performances of the two-level algorithm is observed, unless the number of subregions is large.
- Theorem 1 seems to be valid for this particular choice of non convex domain, even if the constants are found to be larger than the ones in the convex case.

Finally, we have tested algorithms (i) and (ii) for different values of η . We recall that η is proportional to the time step of the FD scheme employed for the approximation of (1). According to the analysis in [7] the condition number of algorithm (ii) can be bounded independently of η . For methods (i) and (ii), Figures 1 and 2 show the estimated condition numbers as functions of η , for different values of the overlap and 8 subregions. Figures 3 and 4 show the results for 64 subdomains.

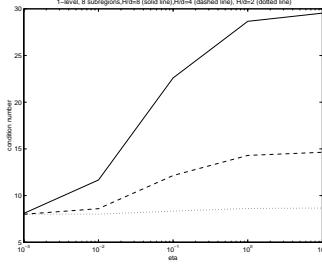


FIGURE 1. Estimated condition number versus η ($H/\delta = 8$, solid line; $H/\delta = 4$, dashed line; $H/\delta = 2$, dotted line): additive 1-level algorithm, $\Omega = (0, 1)^3$, $16 \times 16 \times 16$ -element fine mesh, 8 subregions.

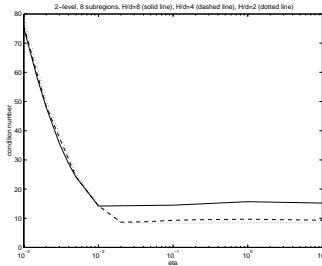


FIGURE 2. Estimated condition number versus η ($H/\delta = 8$, solid line; $H/\delta = 4$, dashed line; $H/\delta = 2$, dotted line): additive 2-level algorithm, $\Omega = (0, 1)^3$, $16 \times 16 \times 16$ -element fine mesh, 8 subregions.

For values of η larger than 0.1, the same remarks made for the case $\eta = 1$ hold. But, in practice, for smaller values of η , the performances of the two-level method deteriorate and the algorithm becomes inefficient for very small η . The coarse space correction is not effective if $\eta < c(\delta) \delta^2$, where $c(\delta)$ is found to be close to one. On the contrary the condition number of method (i) is found to be decreasing with the time step and, as η tends to zero, tends to N_c . For $\eta < c(\delta) \delta^2$, the latter algorithm has better performances than the two-level one, or, in other words, for a fixed value of the time step, the overlap should not be too large or the coarse space correction will not be effective.

Acknowledgments

The author is grateful to Olof Widlund for his endless help and enlightening discussions of my work.

References

1. Franck Assous, Pierre Degond, Ernst Heintze, Pierre-Arnaud Raviart, and J. Segre, *On a finite element method for solving 3D Maxwell equations*, J. Comput. Phys. **109** (1993), 222–237.
2. Satish Balay, William Gropp, Lois Curfman McInnes, and Barry F. Smith, *PETSc Users' Manual*, Tech. Report ANL-95/11, Mathematics and Computer Science Division, Argonne National Laboratory, 1995.
3. Robert Dautray and Jaques-Louis Lions, *Mathematical analysis and numerical methods for science and technology*, Springer-Verlag, New York, 1988.

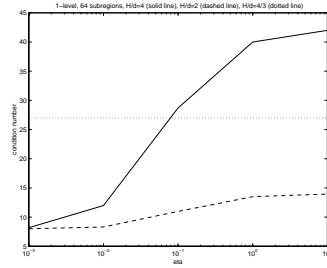


FIGURE 3. Estimated condition number versus η ($H/\delta = 4$, solid line; $H/\delta = 2$, dashed line; $H/\delta = 4/3$, dotted line): additive 1-level algorithm, $\Omega = (0, 1)^3$, $16 \times 16 \times 16$ -element fine mesh, 64 subregions.

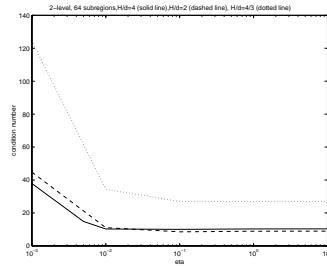


FIGURE 4. Estimated condition number versus η ($H/\delta = 4$, solid line; $H/\delta = 2$, dashed line; $H/\delta = 4/3$, dotted line): additive 2-level algorithm, $\Omega = (0, 1)^3$, $16 \times 16 \times 16$ -element fine mesh, 64 subregions.

4. Maksymilian Dryja and Olof B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), no. 3, 604–620.
5. Vivette Girault and Pierre-Arnaud Raviart, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, New York, 1986.
6. Ralf Hiptmair, *Multigrid method for Maxwell's equations*, Tech. Report 374, Institut für Mathematik, Universität Augsburg, 1997, submitted to SIAM J. Numer.Anal.
7. Ralf Hiptmair and Andrea Toselli, *Overlapping Schwarz methods for vector valued elliptic problems in three dimensions*, Tech. Report 746, Courant Institute for Mathematical Sciences, New York University, October 1997, submitted to the Proceedings of the IMA workshop on ‘Parallel Solution of PDE’, June 1997.
8. Jean-Claude Nédélec, *Mixed finite elements in R^3* , Numer. Math. **35** (1980), 315–341.
9. ———, *A new family of mixed finite elements in R^3* , Numer. Math. **50** (1986), 57–81.
10. Alfio Quarteroni and Alberto Valli, *Numerical approximation of partial differential equations*, Springer-Verlag, Berlin, 1994.
11. Barry F. Smith, Petter Bjørstad, and William Gropp, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, 1996.
12. Andrea Toselli, *Overlapping Schwarz methods for Maxwell's equations in three dimensions*, Tech. Report 736, Courant Institute of Mathematical Sciences, New York University, June 1997, submitted to Numer. Math.

COURANT INSTITUTE OF MATHEMATICAL SCIENCES, 251 MERCER ST, NEW YORK, NY 10012
E-mail address: `toselli@cims.nyu.edu`

Parallel Computing for Reacting Flows Using Adaptive Grid Refinement

Robbert L. Verweij, Aris Twerda, and Tim W.J. Peeters

1. Introduction

The paper reports on the parallelisation of the computational code for modelling turbulent combustion in large industrial 3D glass melting furnaces. Domain decomposition is used to perform the parallelisation. Performance optimisation is examined, to both speed up the code as well as to improve convergence. Local grid refinement is discussed using several different refinement criteria.

The numerical simulation of turbulent reacting flows requires advanced models of turbulence, combustion and radiation in conjunction with sufficiently fine numerical grids to resolve important small scale interactions in the areas of flame front, high shear and near solid walls. These simulations are very CPU- and memory-demanding. Currently, many of the numerical simulation have to be performed with relatively simple models, hampering an accurate prediction. Parallel processing is regarded nowadays as the promising route by which to achieve desired accuracy with acceptable turn-around time.

Domain decomposition is very suitable technique to achieve a parallel algorithm. Furthermore, it allows block-structured refinement quite easily. In this manner the number of grid points can be minimised, giving rise to a very efficient distribution of grid points over the domain.

2. Physical model

Figure 1 shows a typical furnace geometry, where the preheated air ($T = 1400$ K, $v = 9$ m/s) and the gas ($T = 300$ K, $v = 125$ m/s) enter the furnace separately. The turbulence, mainly occurring because of the high gas-inlet velocity, leads to good turbulent mixing. This mixing is essential for combustion of the initially non-premixed fuel and oxidiser into products, which exit the furnace at the opposite side.

The maximum time-averaged temperatures encountered in the furnace are typically 2200 K, at which temperature most of the heat transfer to the walls is by radiation.

The implementation involves the solving of the 3D incompressible (variable density) stationary conservation laws for mass, momentum, energy and species. The hydrodynamic and caloric equations of state are added to relate the pressure

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65Y05, 76T05.

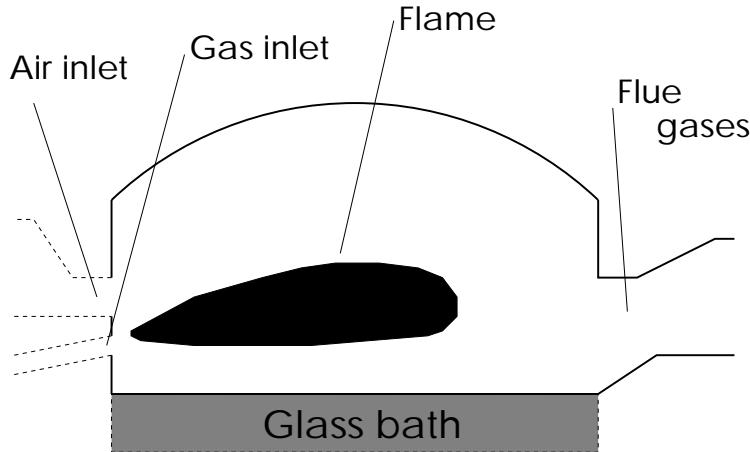


FIGURE 1. Artist's impression of a furnace geometry with flame.

to the density and the enthalpy to the temperature. The equations are averaged using Favre-averaging. The mean flow equations are closed using the standard high-Reynolds $k - \varepsilon$ turbulence model with standard values for all constants. Wall functions are used to bridge the low-Reynolds region near the walls. The heat transfer to the walls is not too much influenced by the velocity field in the near-wall region, justifying this assumption. The conserved-scalar approach is used to model the combustion. The chemistry is modelled with a constrained-equilibrium assumption. An assumed shape β Probability Density Functions (PDF) is used to obtain the mean values of the thermochemical quantities. Radiative heat transfer in the furnace is calculated using the Discrete Transfer Model. A more detailed description of turbulent combustion modelling for furnaces can be found in [1, 6].

3. Numerical model

The modelled equations are discretised using the Finite Volume Method on a colocated, Cartesian grid [4]. The SIMPLE-algorithm is applied to couple the pressure and velocity fields and satisfy mass- and momentum conservation. The linearised systems are solved using the SIP-algorithm or the GMRES-algorithm. The convective terms are discretised using central discretisation, as suggested by [4], but other convective schemes (upwind, tvd) are also available. Full multi-grid [4] is used to improve the convergence behaviour of the multi-block code.

For the domain decomposition the grid-embedding technique [3] is used. This means that one global (coarse) grid is defined, and the domains are defined as subdomains of this coarse grid, where minimal overlap between the blocks is used. Every subdomain can be refined with respect to the coarse grid, with the restriction that adjacent blocks can only be equally fine, twice as fine or twice as coarse as the neighbouring blocks. This is not really a restriction, since the truncation error over the interface is of the order of the size of the coarser block, so patching a very fine block to a very coarse block would not lead to better numerical accuracy. A

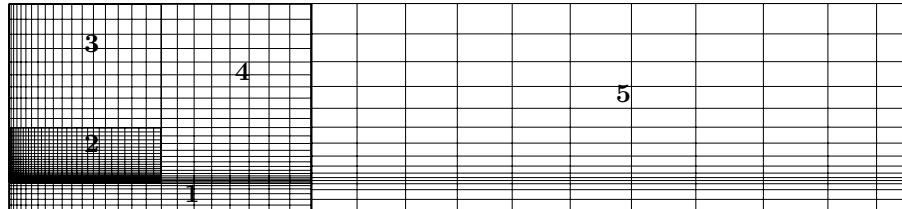


FIGURE 2. Two-dimensional view of a domain decomposition into five blocks with three different levels of grid refinement. For clarity the overlap areas between the blocks have not been drawn.

2D slice of a typical domain decomposition with local grid refinement is shown in Figure 2.

To couple the domains, one layer of halo-cells (auxiliary control volumes) was used [7]. The values on the internal boundaries are copied into the halo-cells of the neighbouring domains. This way of coupling renders the need for explicit boundary conditions on the internal boundaries superfluous and guarantees that the converged solution is independent of the block decomposition, if all blocks are equally fine.

The local grid refinement was implemented by computing the fluxes on the fine grid side of the interfaces only, and then sending them to the coarser grids and adding them there. This yields a flux conservative scheme over the block-interface. For the other quantities, tri-linear interpolation and splines are used. This also uniquely defines the value of the gradients on the interfaces.

4. Parallel implementation

The domain decomposition was used as a basis for parallelisation. Static load balancing is performed; every domain contains (approximately) the same amount of grid-points and the amount of work per grid point was assumed to be constant for all points. All processors were assumed to be equally fast. In combination with local grid refinement some load-imbalance was accepted to minimise the number of blocks and optimise the number of grid points needed.

Exactly one domain is computed per processor, so that the (old) sequential single-block program could easily be used for multi-block computations. The SPMD (Single Program, Multiple data) programming model was used. Typically 4 to 32 domains were used to perform the calculations.

In this study, PVM and MPI were adopted on clustered workstations and the machine specific message passing tool SHMEM on the CRAY T3E. Changing from one message passing interface to another proved easy, since all communication was hidden in a few generic subroutines, named for example *parsend* or *parrecv*. In domain decomposition the number and size of messages is relatively small compared to the computations done. Hence switching from MPI to SHMEM didn't yield significantly better results.

5. Results

5.1. Performance optimisation. First the speedup of the code was examined. The 3D version of the code, called FURNACE is shown in Figure 3 on the left for two grid-sizes (20^3 and 50^3). On the right the speedup for a 2D version of the

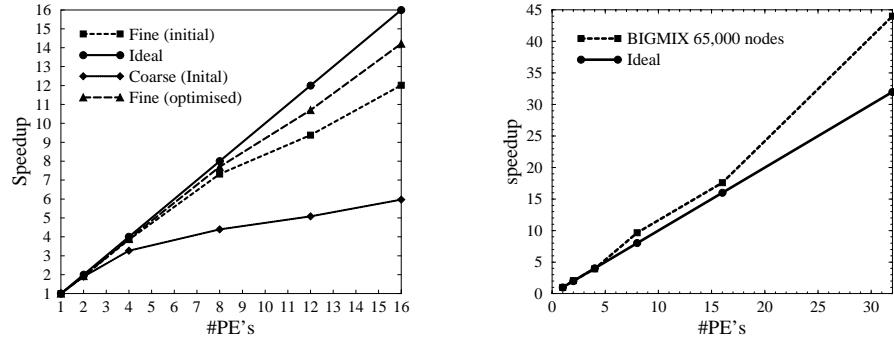


FIGURE 3. Speedup for several turbulent combustion codes **left:** 3D **right:** 2D

code, called BIGMIX, with more elaborate chemistry [6] is shown. All test-runs have been performed on a CRAY T3E AC80-128.

Note that FURNACE scales better for finer grids, as is expected. The super-linear speedup of BIGMIX can be explained by assuming better cache-use, because of the two dimensionality of the arrays. Triggered by the sensitivity of the code performance on the cache-use, the original code was slightly rewritten, and optimised to enable better vector-length controlling. Compiler-options were investigated and it was found out that `-O3,Unroll12,nojump` together with the enabling of the CRAY T3E data-stream buffers significantly improved the single PE performance as well as the speedup, since there was less load-imbalance. This is shown in Figure 3 (left) too.

The effect of load imbalance can also be seen in the weird bend in the speedup curve in the left figure around 8 and 12 PE's, where speedup seems to decrease and then increase again, especially for the coarse grid. This bend has been analysed by plotting the time spent in different parts of FURNACE. It can be seen from Figure 4 that when using more blocks the amount of time spent in updating the internal boundaries (which requires communication) becomes the bottleneck. This is probably due to the fact that if there are more than 8 blocks in the decomposition, some blocks have more neighbours than others, which implies that some blocks will update more boundaries than others. This assumption is asserted in the next paragraph.

Further time-reduction was obtained by minimising the amount of I/O requests. I/O is very expensive and still highly sequential on parallel machines (although recently MPI-2 defines a standard to optimise this, which has not been used here). In general it was found that if a data-block needs to be read in by all processors, it is the quickest to let 1 PE read the data and then global scatter it to all other PE's.

TABLE 1. Load imbalance for complete problem

Time needed (s)	sliced decomp.	rectangular decomp.
Total program time	2150	1700
Parallel working time	1140 (53%)	1241 (73%)
Total barrier waiting time	606 (28%)	106 (6%)
Total communication time	85 (4%)	77 (5%)

TABLE 2. Relative change of some variables with respect to their finest grid values

Grid size	heat flux	flue temp.	flue O ₂ conc.	flue NO _x conc.
16 × 24 × 20	+ 17%	+ 2%	+ 2%	-33%
24 × 36 × 30	+ 12%	+ 3%	+ 1%	-25%
32 × 48 × 40	+ 8%	+ 3%	+ 1%	-12%
32 × 72 × 60	+ 0%	+ 0%	+ 0%	+ 0%

In the original code every variable could be read from file independently, causing much I/O interrupts. In the modified code, all data was available in a single datafile, and the amount of file checking was minimised. The total amount of I/O time for both methods is depicted in Figure 4 for a run on 16 PE’s, the total amount of data to be read from disk is $16 \times 5.4\text{Mb}$ (this includes the lookup table for chemistry) The amount of data written to disk is $16 \times 1.4\text{Mb}$. The differences per PE are much smaller in the modified code, and the mean time spent in I/O was brought down from 197.1 seconds to 15.4 seconds.

The assumption that load imbalance was caused by the fact that some blocks have more neighbours than others was asserted next. 200 iterations were done on a coarse ($16 \times 24 \times 20$) grid, used in previous studies [1, 6]. This grid was split into four blocks in two different configurations, shown in Figure 5. In the left decomposition all blocks have exactly two neighbours (the rectangular decomposition), in the right one the middle two blocks have two neighbours (the sliced decomposition), and the outer blocks one. Every block contains the same amount of points. Both decompositions lead to exactly the same converged solution, but the timings are quite different, as shown in Table 1.

The rectangular decomposition yields much better timing results than the sliced decomposition. This confirms the assumption that in the sliced decomposition, the two middle blocks have much more communications to do, creating an huge load-imbalance in that part of the code. This effect becomes smaller if the number of points per blocks becomes bigger. Note that the total communication time is approximately equal in both cases, and is small compared to the entire program time. The smaller parallel working time in the rectangular decomposition is because of better cache use, we assume. The load imbalance becomes smaller if the number of points per block become larger, as could be seen in Figure 4.

5.2. Local grid refinement. The block-structured approach allows for local grid refinement in each block, as was already explained in the previous section. The need for local grid refinement becomes clear from Table 2, where the relative change of some variables with respect to the value of the variable on the finest grid is printed, if the global grid is varied from relatively coarse to very fine.

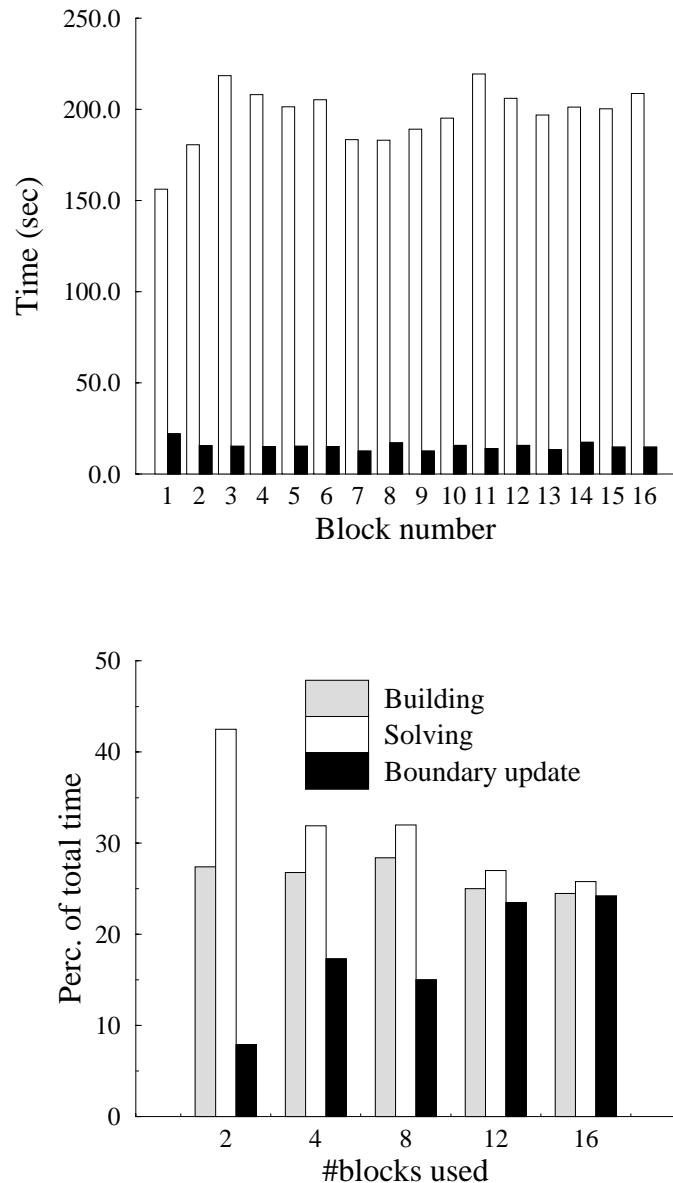


FIGURE 4. **top:** Total time spent in I/O for each PE **bottom:** Percentage of time spent in different parts of the code for different number of blocks

Although the temperature and the heat flux are not so much influenced by finer grids, the predictions for the concentrations show a great grid sensitivity, mainly because they depend heavily on the mixture fraction concentration prediction, which also varies considerably.

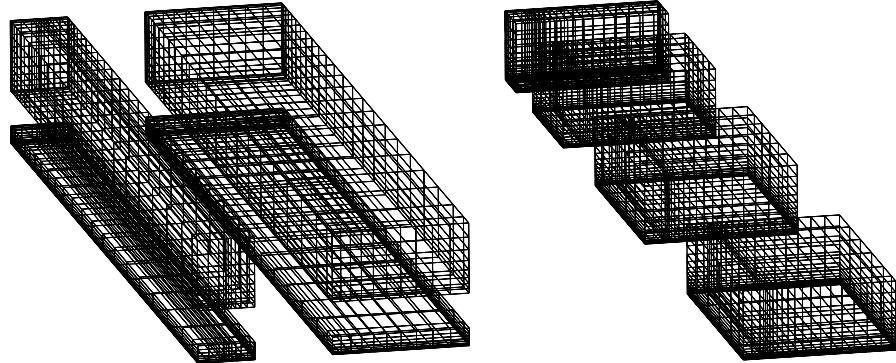


FIGURE 5. Two block decompositions of the same grid. **left:** Rectangular decomposition **right:** Sliced decomposition.

Also another, standard, testcase was performed: The laminar lid-driven cavity at $Re = 100$. This testcase was also used by [5] to test his dynamic local grid refinement. Also here grid independent results were only obtained at a 256^2 grid, in agreement with [8], showing the amount of grid points needed without local refinement.

Statrical grid refinement was applied first; the blocks where manually generated, based on experience and intuition rather than some fundamental refinement criterion. This did yield satisfactory results, in the sense that first results yielded reduction of grid points from 38 % for the entire combustion computations [1] to 80 % for the cavity.

Next, dynamical grid refinement was implemented. Some error estimator should provide information about the refinement. In literature, the Richardson extrapolation is often used as criterion for refinement. However, for combustion problems, where geometry and physics are complex, the coarsest reliable mesh contains so many grid points that a uniform refinement, just to estimate the error, cannot be afforded. Muzaferia [5] developed a method, similar to Richardson extrapolation, which is based on the difference of higher- and lower order approximation of the fluxes to approximate the truncation error. Apart from this method, a more physical criterion was build in, to reflect our believe that in complex flows steep gradients of relevant quantities also mark regions which should be refined [2]. Hence the gradient of an 'interesting quantity' is used as an error indication. In the furnace-simulation another interesting quantity like temperature or density could be used. However, in most flows it is not the gradient, but the curvature (the second derivative) which yields the areas of high shear. These three criteria have been implemented and tested on several flows.

To test the different criteria the laminar lid-driven cavity was computed, to compare the results to [5], and since this problem converges very quickly at low Reynolds numbers. Grid independent results were only obtained at a 256^2 grid, in agreement with [8], showing the amount of grid points that are needed without local refinement. The results are in excellent agreement with those of [5] and [8]. However, switching to other criteria gave different results. Figure 6 shows the

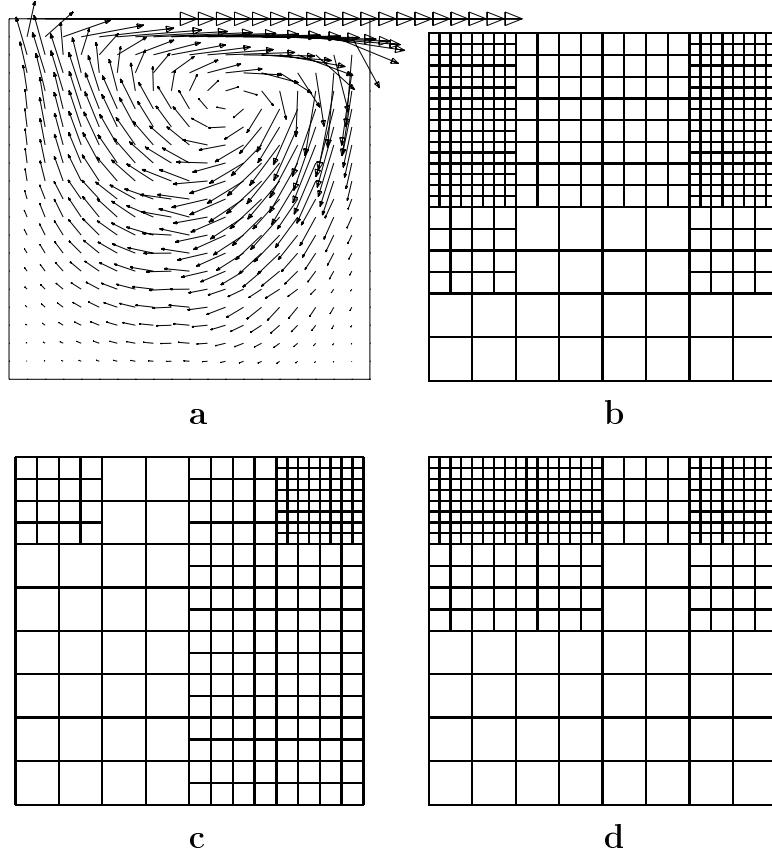


FIGURE 6. Grids, obtained with different local refinement criteria
a Vectorplot of field **b** Maximum curvature **c** Method of [5] and present method **d** Maximum velocity-gradient criterion

different grid refinements after 2 levels of refinement starting with a uniform 8^2 grid, and allowing 4 blocks in both directions.

The results for refinement criterion c (the Muzaferia method) are in excellent agreement with those of [5] and [8]. All three criteria yield plausible refinement regions at first view and comparable minimum values for the streamlines. More testcases need to be considered to determine which criterion yields the most accurate results by comparing them to the 'true' solution, ie. the grid independent solution.

A backdraw of the block-structured refinement is that it leads to severe load imbalance, as shown in Table 3. This could be overcome by using a different way of defining the new domain decomposition. Currently the decomposition is based on a minimal number of blocks. Other options might be to use many very small blocks (although this will probably influence the convergence behaviour) or to run several blocks on 1 PE, which is currently not possible.

TABLE 3. Load imbalance due to local grid refinement.

Criterion	# blocks	grid points		
		Max	Min	Total
Muzaferia	6	128	4	220
Gradient	7	256	4	292
Curvature	7	128	8	376

6. Conclusions

Load imbalance is a underestimated problem in most CFD applications. Parallelisation by domain decomposition is a straightforward and efficient way to parallelise combustion codes. When combined with multigrid, convergence becomes independent of the number of blocks used. Good tuning of a program significantly improves the performance on cache-based machines like the CRAY T3E.

Dynamical grid refinement yields significant reduction of grid points with the same accuracy. This is needed since tests showed the need of huge grids for realistic applications. Different criteria can be applied to determine the refinement, yielding different regions of refinement. The choice for the 'best' criterion is not obvious yet, and might even depend on the application.

References

1. G.P. Boerstoel, *Modelling of gas-fired furnaces*, Ph.D. thesis, TU Delft, may 1997.
2. W.L. Chen, F.S. Lien, and M.A. Leschziner, *Local mesh refinement within a multi-block structured grid scheme for general flows*, Comp. Meth. Appl. Mech. Eng. **144** (1997), no. 3, 327–327.
3. P. Coelho, J.C.F. Pereira, and M.G. Carvalho, *Calculation of laminar recirculating flows using a local non-staggered grid refinement system.*, Int. J. Num. Meth. Fl. **12** (1991), 535–557.
4. J.H. Ferziger and M Perić, *Computational methods for fluid dynamics*, Springer-Verlag, Berlin, 1996.
5. S. Muzaferia and D. Gosman, *An adaptive finite-volume discretisation technique for unstructured grids*, Tech. Report Ms. No. G0353, Dept. of Mech. Eng., Imperial College of Science, Technology and Medicine, 1997.
6. T.W.J. Peeters, *Numerical modeling of turbulent natural-gas diffusion flames.*, Ph.D. thesis, TU Delft, September 1995.
7. M. Perić, M. Schäfer, and E. Schreck, *Computation of fluid flow with a parallel multigrid solver.*, Parallel Computational Fluid Dynamics 1991, Elsevier Science Publishers. B.V., 1992, pp. 297–312.
8. M.C. Thompson and J.H. Ferziger, *An adaptive multigrid technique for the incompressible navier Stokes equations*, Journ. Comp. Phys. **82** (1989), 94–121.

DEPARTMENT OF APPLIED PHYSICS, DELFT UNIVERSITY OF TECHNOLOGY, LORENTZWEG 1,
2628 CJ DELFT, THE NETHERLANDS
E-mail address: R.Verweij@tn.tudelft.nl

DEPARTMENT OF APPLIED PHYSICS, DELFT UNIVERSITY OF TECHNOLOGY, LORENTZWEG 1,
2628 CJ DELFT, THE NETHERLANDS
E-mail address: A.Twerda@tn.tudelft.nl

DEPARTMENT OF APPLIED PHYSICS, DELFT UNIVERSITY OF TECHNOLOGY, LORENTZWEG 1,
2628 CJ DELFT, THE NETHERLANDS
E-mail address: T.W.J.Peeters@tn.tudelft.nl

The Coupling of Mixed and Conforming Finite Element Discretizations

Christian Wieners and Barbara I. Wohlmuth

1. Introduction

In this paper, we introduce and analyze a special mortar finite element method. We restrict ourselves to the case of two disjoint subdomains, and use Raviart-Thomas finite elements in one subdomain and conforming finite elements in the other. In particular, this might be interesting for the coupling of different models and materials. Because of the different role of Dirichlet and Neumann boundary conditions a variational formulation without a Lagrange multiplier can be presented. It can be shown that no matching conditions for the discrete finite element spaces are necessary at the interface. Using static condensation, a coupling of conforming finite elements and enriched nonconforming Crouzeix-Raviart elements satisfying Dirichlet boundary conditions at the interface is obtained. Then the Dirichlet problem is extended to a variational problem on the whole nonconforming ansatz space. In this step a piecewise constant Lagrange multiplier comes into play. By eliminating the local cubic bubble functions, it can be shown that this is equivalent to a standard mortar coupling between conforming and Crouzeix-Raviart finite elements. Here the Lagrange multiplier lives on the side of the Crouzeix-Raviart elements. And in contrast to the standard mortar P1/P1 coupling the discrete ansatz space for the Lagrange multiplier consists of piecewise constant functions instead of continuous piecewise linear functions. We note that the piecewise constant Lagrange multiplier represents an approximation of the Neumann boundary condition at the interface. Finally, we present some numerical results and sketch the ideas of the algorithm. The arising saddle point problems are solved by multigrid techniques with transforming smoothers.

The mortar methods have been introduced recently and a lot of work in this field has been done during the last few years; cf., e.g., [1, 4, 5, 14, 15]. For the construction of efficient iterative solvers we refer to [2, 3, 20, 21]. The concepts of a posteriori error estimators and adaptive refinement techniques have also been generalized to mortar methods on nonmatching grids; see e.g. [13, 22, 25, 24].

1991 *Mathematics Subject Classification*. Primary 65N15; Secondary 65N30, 65N55.

Key words and phrases. mortar finite elements, mixed finite elements, multigrid methods.

This work was supported in part by the Deutsche Forschungsgemeinschaft.

Originally introduced for the coupling of spectral element methods and finite elements, this method has thus now been extended to a variety of special situations [6, 7, 11, 12, 26].

2. The continuous problem

We consider the following elliptic boundary value problem

$$(1) \quad \begin{aligned} Lu := -\operatorname{div}(a\nabla u) + bu &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma := \partial\Omega \end{aligned}$$

where Ω is a bounded, polygonal domain in \mathbb{R}^2 and $f \in L^2(\Omega)$. Furthermore, we assume $a = (a_{ij})_{i,j=1}^2$ to be a symmetric, uniformly positive definite matrix-valued function with $a_{ij} \in L^\infty(\Omega)$, $1 \leq i, j \leq 2$, and $0 \leq b \in L^\infty(\Omega)$. The domain $\overline{\Omega}$ is decomposed into two nonoverlapping polyhedral subdomains $\overline{\Omega}_1 \cup \overline{\Omega}_2$, and we assume that $\operatorname{meas}(\partial\Omega_2 \cap \partial\Omega) \neq 0$. On Ω_1 we introduce a mixed formulation of the elliptic boundary value problem (1) with a Dirichlet boundary condition on $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$, whereas on Ω_2 we use the standard variational formulation with a Neumann boundary condition on Γ . We denote by \mathbf{n} the outer unit normal of Ω_1 . The Dirichlet boundary condition on Γ will be given by the weak solution u_2 in Ω_2 and the Neumann boundary condition by the flux \mathbf{j}_1 in Ω_1 . Then, the ansatz space for the solution (\mathbf{j}_1, u_1) in Ω_1 is given by $H(\operatorname{div}; \Omega_1) \times L^2(\Omega_1)$ and by $H_{0;\Gamma_2}^1(\Omega_2) := \{v \in H^1(\Omega_2) \mid v|_{\Gamma_2} = 0\}$, where $\Gamma_2 := \partial\Omega \cap \partial\Omega_2$ for the solution u_2 in Ω_2 . We recall that no boundary condition on Γ has to be imposed on the ansatz spaces. In contrast to the standard case, Neumann boundary conditions are essential boundary conditions for the mixed formulation, i.e. they have to be enforced in the construction of the ansatz spaces. The coupling of the mixed and standard formulations leads to the following saddle point problem:

Find $(\mathbf{j}_1, u_1, u_2) \in H(\operatorname{div}; \Omega_1) \times L^2(\Omega_1) \times H_{0;\Gamma_2}^1(\Omega_2)$ such that

$$(2) \quad \begin{aligned} a_1(\mathbf{j}_1, \mathbf{q}_1) + b(\mathbf{q}_1, u_1) - d(\mathbf{q}_1, u_2) &= 0, & \mathbf{q}_1 \in H(\operatorname{div}; \Omega_1), \\ b(\mathbf{j}_1, v_1) - c(u_1, v_1) &= -(f, v_1)_{0;\Omega_1}, & v_1 \in L^2(\Omega_1), \\ -d(\mathbf{j}_1, v_2) - a_2(u_2, v_2) &= -(f, v_2)_{0;\Omega_2}, & v_2 \in H_{0;\Gamma_2}^1(\Omega_2). \end{aligned}$$

Here the bilinear forms $a_i(\cdot, \cdot)$, $1 \leq i \leq 2$, $b(\cdot, \cdot)$, $c(\cdot, \cdot)$ and $d(\cdot, \cdot)$ are given by

$$\begin{aligned} a_2(w_2, v_2) &:= \int_{\Omega_2} (a \nabla v_2 \cdot \nabla w_2 + b v_2 w_2) dx, & v_2, w_2 \in H_{0;\Gamma_2}^1(\Omega_2), \\ a_1(\mathbf{p}_1, \mathbf{q}_1) &:= \int_{\Omega_1} a^{-1} \mathbf{p}_1 \cdot \mathbf{q}_1 dx, & \mathbf{p}_1, \mathbf{q}_1 \in H(\operatorname{div}; \Omega_1), \\ b(\mathbf{q}_1, v_1) &:= \int_{\Omega_1} \operatorname{div} \mathbf{q}_1 v_1 dx, & v_1 \in L^2(\Omega_1), \mathbf{q}_1 \in H(\operatorname{div}; \Omega_1), \\ c(w_1, v_1) &:= \int_{\Omega_1} b w_1 v_1 dx, & v_1, w_1 \in L^2(\Omega_1), \\ d(\mathbf{q}_1, v_2) &:= \langle \mathbf{q}_1 \mathbf{n}, v_2 \rangle, & \mathbf{q}_1 \in H(\operatorname{div}; \Omega_1), v_2 \in H_{0;\Gamma_2}^1(\Omega_2), \end{aligned}$$

and $\langle \cdot, \cdot \rangle$ stands for the duality pairing of $H^{-1/2}(\Gamma)$ and $H^{1/2}(\Gamma)$. The kernel of the operator $B : H(\operatorname{div}; \Omega_1) \times H_{0;\Gamma_2}^1(\Omega_2) \rightarrow L^2(\Omega_1)$, which is associated with the linear form $b(\cdot, v_1)$ is $\operatorname{Ker}B := \{(\mathbf{q}_1, v_2) \in H(\operatorname{div}; \Omega_1) \times H_{0;\Gamma_2}^1(\Omega_2) \mid \operatorname{div} \mathbf{q} = 0\}$. On $H(\operatorname{div}; \Omega_1) \times H_{0;\Gamma_2}^1(\Omega_2)$, we introduce the nonsymmetric bilinear form $a(\sigma, \tau) := a_2(w_2, v_2) + d(\mathbf{p}_1, v_2) + a_1(\mathbf{p}_1, \mathbf{q}_1) - d(\mathbf{q}_1, w_2)$ where $\sigma := (\mathbf{q}_1, v_2)$, $\tau := (\mathbf{p}_1, w_2)$, and the norm $\|\cdot\|$ is given by $\|\sigma\|^2 := \|v_2\|_{1;\Omega_2}^2 + \|\mathbf{q}_1\|_{\operatorname{div}; \Omega_1}^2$. Taking the continuity of the bilinear forms, the Babuška-Brezzi condition, and the coercivity of $a(\cdot, \cdot)$ on $\operatorname{Ker}B$, $a(\sigma, \sigma) \geq \alpha \|\sigma\|^2$, $\sigma \in \operatorname{Ker}B$, into account, we obtain unique solvability of the saddle point problem (2); see e.g. [19].

3. Discretization and A Priori Estimates

We restrict ourselves to the case that simplicial triangulations \mathcal{T}_{h_1} and \mathcal{T}_{h_2} are given on both subdomains Ω_1 and Ω_2 . However, our results can be easily extended to more general situations including polar grids. The sets of edges of the meshes are denoted by \mathcal{E}_{h_1} and \mathcal{E}_{h_2} . We use the Raviart-Thomas space of order k_1 , $RT_{k_1}(\Omega_1; \mathcal{T}_{h_1}) \subset H(\text{div}; \Omega_1)$, $k_1 \geq 0$, for the approximation of the flux \mathbf{j}_1 in Ω_1 , the space of piecewise polynomials of order k_1 , $W_{k_1}(\Omega_1; \mathcal{T}_{h_1}) := \{v \in L^2(\Omega_1) \mid v|_T \in P_{k_1}(T), T \in \mathcal{T}_{h_1}\}$ for the approximation of the primal variable u_1 in Ω_1 , and conforming P_{k_2} finite elements $S_{k_2}(\Omega_2; \mathcal{T}_{h_2}) \subset H_{0;\Gamma_2}^1(\Omega_2)$ in Ω_2 . Associated with this discretization is the following discrete saddle point problem:

Find $(\mathbf{j}_{h_1}, u_{h_1}, u_{h_2}) \in RT_{k_1}(\Omega_1; \mathcal{T}_{h_1}) \times W_{k_1}(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})$ such that

(3)

$$\begin{aligned} a_1(\mathbf{j}_{h_1}, \mathbf{q}_h) + b(\mathbf{q}_h, u_{h_1}) - d(\mathbf{q}_h, u_{h_2}) &= 0, & \mathbf{q}_h \in RT_{k_1}(\Omega_1; \mathcal{T}_{h_1}), \\ b(\mathbf{j}_{h_1}, w_h) - c(u_{h_1}, w_h) &= -(f, w_h)_{0;\Omega_1}, & w_h \in W_{k_1}(\Omega_1; \mathcal{T}_{h_1}), \\ -d(\mathbf{j}_{h_1}, v_h) - a_2(u_{h_2}, v_h) &= (f, v_h)_{0;\Omega_2}, & v_h \in S_{k_2}(\Omega_2; \mathcal{T}_{h_2}). \end{aligned}$$

It can be easily seen that the discrete Babuška-Brezzi condition is satisfied with a constant independent of the refinement level. In addition, the kernel of the discrete operator B_{k_1} is a subspace of $\text{Ker } B$. Therefore, an upper bound for the discretization error is given by the best approximation, and we obtain the well known a priori estimate, see e.g. [19],

$$(4) \quad \begin{aligned} & \| \mathbf{j} - \mathbf{j}_{h_1} \|_{\text{div}; \Omega_1}^2 + \| u - u_{h_1} \|_{0; \Omega_1}^2 + \| u - u_{h_2} \|_{1; \Omega_2}^2 \\ & \leq C \left(h_1^{2(k_1+1)} (\| u \|_{k_1+1; \Omega_1}^2 + \| \mathbf{j} \|_{k_1+1; \Omega_1}^2 + \| f \|_{k_1+1; \Omega_1}^2) + h_2^{2k_2} \| u \|_{k_2+1; \Omega_2}^2 \right) \end{aligned}$$

if the problem has a regular enough solution. In fact, the constant C is independent of the ratio of h_1 and h_2 and there is no matching condition for the triangulations \mathcal{T}_{h_1} and \mathcal{T}_{h_2} at the interface required.

4. An Equivalent Nonconforming Formulation

It is well known that mixed finite element techniques are equivalent to nonconforming ones [8]. Introducing interelement Lagrange multipliers, the flux variable as well as the primal variable can be evaluated locally and the resulting Schur complement system is the same as for the positive definite variational problem associated with a nonstandard nonconforming Crouzeix-Raviart discretization [19]. In addition, the mixed finite element solution can be obtained by a local postprocessing from these Crouzeix-Raviart finite element solution.

We now restrict ourselves to the lowest order Raviart-Thomas ansatz space ($k_1 = 0$). To obtain the equivalence, we consider the enriched Crouzeix-Raviart space $NC(\Omega_1; \mathcal{T}_{h_1}) := CR(\Omega_1; \mathcal{T}_{h_1}) + B_3(\Omega_1; \mathcal{T}_{h_1})$ where $CR(\Omega_1; \mathcal{T}_{h_1})$ is the Crouzeix-Raviart space of piecewise linear functions which are continuous at the midpoints of the triangulation \mathcal{T}_{h_1} and equal to zero at the midpoints of any boundary edge $e \in \mathcal{E}_{h_1} \cap \partial\Omega$. $B_3(\Omega_1; \mathcal{T}_{h_1})$ is the space of piecewise cubic bubble functions which vanish on the boundary of the elements. Then, we can obtain equivalence between the saddle point problem

$$(5) \quad \begin{aligned} a_1(\mathbf{j}_{h_1}, \mathbf{q}_h) + b(\mathbf{q}_h, u_{h_1}) &= d(\mathbf{q}_h, u_{h_2}), & \mathbf{q}_h \in RT_0(\Omega_1; \mathcal{T}_{h_1}) \\ b(\mathbf{j}_{h_1}, w_h) - c(u_{h_1}, w_h) &= -(f, w_h)_{0;\Omega_1}, & w_h \in W_0(\Omega_1; \mathcal{T}_{h_1}) \end{aligned}$$

and the positive definite problem: Find $\Psi_{h_1} \in NC^{u_{h_2}}(\Omega_1; \mathcal{T}_{h_1})$ such that

$$(6) \quad a_{NC}(\Psi_{h_1}, \psi_h) = (f, \Pi_0 \psi_h)_{0; \Omega_1}, \quad \psi_h \in NC^0(\Omega_1; \mathcal{T}_{h_1}).$$

Here, $a_{NC}(\phi_h, \psi_h) := \sum_{T \in \mathcal{T}_{h_1}} \int_T P_{a^{-1}}(a \nabla \phi_h) \nabla \psi_h + b \Pi_0 \phi_h \Pi_0 \psi_h \, dx$, and Π_0 stands for the L^2 -projection onto $W_0(\Omega_1; \mathcal{T}_{h_1})$. $P_{a^{-1}}$ is the weighted L^2 -projection, with weight a^{-1} , onto the three dimensional local Raviart-Thomas space of lowest order, and $NC^g(\Omega_1; \mathcal{T}_{h_1}) := \{\psi_h \in NC(\Omega_1; \mathcal{T}_{h_1}) \mid \int_e \Psi_{h_1} \, d\sigma = \int_e g \, d\sigma, e \in \mathcal{E}_{h_1} \cap \Gamma\}$.

Using the equivalence of (5) and (6) in (3), we get:

Find $(\Psi_{h_1}, u_{h_2}) \in NC^{u_{h_2}}(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})$ such that

$$(7) \quad \begin{aligned} a_{NC}(\Psi_{h_1}, \psi_h) &= (f, \Pi_0 \psi_h)_{0; \Omega_1}, \quad \psi_h \in NC^0(\Omega_1; \mathcal{T}_{h_1}), \\ a_2(u_{h_2}, v_h) + d(P_{a^{-1}}(a \nabla \Psi_{h_1}), v_h) &= (f, v_h)_{0; \Omega_2}, \quad v_h \in S_{k_2}(\Omega_2; \mathcal{T}_{h_2}). \end{aligned}$$

Note that the ansatz space on Ω_1 depends on the solution in Ω_2 .

For the numerical solution, we transfer (7) into a saddle point problem where no boundary condition has to be imposed on the ansatz spaces at the interface. It can be shown that the Dirichlet problem (6) can be extended to a variational problem on the whole space $NC(\Omega_1; \mathcal{T}_{h_1})$. In fact, we obtain

$$(8) \quad a_{NC}(\Psi_{h_1}, \psi_h) - d(P_{a^{-1}}(a \nabla \Psi_{h_1}), \psi_h) = (f, \Pi_0 \psi_h)_{0; \Omega_1}, \quad \psi_h \in NC(\Omega_1; \mathcal{T}_{h_1}).$$

Let $M(\Gamma; \mathcal{E}_{h_1}) := \{\mu \in L^2(\Gamma) \mid \mu|_e \in P_0(e), e \in \mathcal{E}_{h_1} \cap \Gamma\}$ be the space of piecewise constant Lagrange multipliers associated with the 1D triangulation of Γ inherited from \mathcal{T}_{h_1} . Then, the condition $\Psi_{h_1} \in NC^{u_{h_2}}(\Omega_1; \mathcal{T}_{h_1})$ is nothing else than $\Psi_{h_1} \in NC(\Omega_1; \mathcal{T}_{h_1})$ and

$$(9) \quad \int_{\Gamma} \mu(\Psi_{h_1} - u_{h_2}) \, d\sigma = 0, \quad \mu \in M(\Gamma; \mathcal{E}_{h_1}).$$

THEOREM 1. Let $(\Psi_{h_1}, u_{h_2}) \in NC^{u_{h_2}}(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})$ be the solution of (7). Then, $u_M := (\Psi_{h_1}, u_{h_2})$ and $\lambda_M := P_{a^{-1}}(a \nabla \Psi_{h_1})|_{\Gamma}$ is the unique solution of the following saddle point problem: Find $(u_M, \lambda_M) \in (NC(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})) \times M(\Gamma; \mathcal{E}_{h_1})$ such that

$$(10) \quad \begin{aligned} a(u_M, v) - \hat{d}(\lambda_M, v) &= f(v), \quad v \in NC(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2}), \\ \hat{d}(\mu, u_M) &= 0, \quad \mu \in M(\Gamma; \mathcal{E}_{h_1}). \end{aligned}$$

Here the bilinear and linear forms are given by:

$$\begin{aligned} a(w, v) &:= a_2(w, v) + a_{NC}(w, v), \quad v, w \in NC(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2}), \\ \hat{d}(\mu, v) &:= \int_{\Gamma} \mu(v|_{\Omega_1} - v|_{\Omega_2}) \, d\sigma, \quad \mu \in M(\Gamma; \mathcal{E}_{h_1}), \\ f(v) &:= (f, v)_{0; \Omega_2} + (f, \Pi_0 v)_{0; \Omega_1}. \end{aligned}$$

Taking (8) and (9) into account, the assertion is an easy consequence of (7).

Theorem 1 states the equivalence of (3) and (10) in the case $k_1 = 0$ with $\mathbf{j}_{h_1} = P_{a^{-1}}(a \nabla u_M|_{\Omega_1})$, $u_{h_1} = \Pi_0 u_M|_{\Omega_1}$ and $u_{h_2} = u_M|_{\Omega_2}$. In fact, (10) is a mortar finite element coupling between the conforming and nonconforming ansatz spaces. The Lagrange multiplier $\lambda_M = \mathbf{j}_{h_1} \mathbf{n}|_{\Gamma}$ is associated with the side of the nonconforming discretization, and it gives an approximation of the Neumann boundary condition on the interface Γ .

REMARK 2. For the numerical solution, we will eliminate locally the cubic bubble functions in (10). In particular, for the special case $b = 0$ and the diffusion coefficient a is piecewise constant, we obtain the standard variational problem for

Crouzeix-Raviart elements where the right hand side f is replaced by $\Pi_0 f$. Then, the nonconforming solution Ψ_{h_1} is given by

$$\Psi_{h_1}|_T = u_{h_1}|_T + \frac{5}{12} \sum_{i=1}^3 h_{e_i}^2 \Pi_0 f|_T (\lambda_1 \lambda_2 \lambda_3), \quad T \in \mathcal{T}_{h_1}$$

where λ_i , $1 \leq i \leq 3$ are the barycentric coordinates, and h_{e_i} is the length of the edge $e_i \subset \partial T$, $1 \leq i \leq 3$. Here, u_{h_1} stands for the Crouzeix-Raviart part of the mortar finite element solution of (10) restricted on $(CR(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})) \times M(\Gamma; \mathcal{E}_{h_1})$.

5. Numerical algorithm

The numerical approximation of (2) is based on the equivalence between mixed and nonconforming finite elements. Thus, we use the variational problem given in Theorem 1, where additional Lagrange multipliers at the interface are required. We recall that the cubic bubble functions in the saddle point problem (10) can be locally eliminated. Thus, we obtain finally a reduced saddle point problem defined on $(CR(\Omega_1; \mathcal{T}_{h_1}) \times S_{k_2}(\Omega_2; \mathcal{T}_{h_2})) \times M(\Gamma; \mathcal{E}_{h_1})$.

The construction of efficient iterative solvers for this type of saddle point problems has often been based on domain decomposition ideas; see e. g. [2, 3, 21]. This approach involves a preconditioner for the exact Schur complement. Here, we apply standard multigrid methods with transforming smoothers. The analysis of transforming smoothers for mortar finite elements is similar to the analysis for the Stokes problem given in [16, 23]. The technical details for the mortar case will be presented in a forthcoming paper.

In contrast to the Stokes problem, the Schur complement for mortar elements is of smaller dimension. It is associated with the one dimensional interface. Thus the Schur complement of the constructed smoother can be assembled exactly without loosing the optimal complexity of the algorithm. In addition, the condition number of the Schur complement of the smoother is bounded independent of the meshsize. And our numerical results indicate that optimal order convergence also can be obtained with an approximate Schur complement.

We present two numerical examples implemented using the software toolbox UG [9, 10] and its finite element library. Two different model problems are considered. We consider $-\operatorname{div}(a \nabla u) = 1$ on $(0; 2) \times (0; 1)$ with homogeneous Dirichlet boundary conditions and a discontinuous coefficient a . On subdomain $\Omega_2 := (0; 1) \times (0; 1)$, a is equal to 1, and on subdomain $\Omega_1 := (1; 2) \times (0; 1)$, a is equal to 0.001. This example shows the effect of nonmatching grids with different stepsizes and a piecewise constant discontinuous diffusion coefficient. In addition, the triangulation on Ω_2 is slightly distorted. Whereas the second example is a simple model for a rotating geometry with two circles which can occur for time dependent problems. Here, we solve $-\Delta u = \sin(x) + \exp(y)$ with homogeneous Dirichlet boundary conditions on the unit circle, and Ω_1 is the interior circle (see Figure 1).

To apply multigrid algorithms to mortar finite elements, we consider a hierarchy $X_0, X_1, X_2, \dots, X_k$ of finite element spaces, where

$$X_l := CR(\Omega_1, \mathcal{T}_{h_1^{(l)}}) \times S_1(\Omega_2, \mathcal{T}_{h_2^{(l)}}) \times M(\Gamma; \mathcal{E}_{h_1^{(l)}})$$

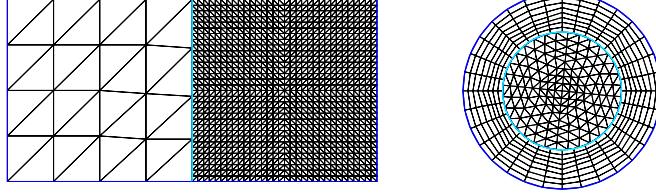


FIGURE 1. Triangulation for example 1 (left) and example 2 (right)

and $h_1^{(l-1)} = 2h_1^{(l)}$, $h_2^{(l-1)} = 2h_2^{(l)}$. Note, that the finite element spaces are nonnested in the first component.

The fine grid problem is of the form

$$(11) \quad K_k z_k = f_k, \quad K_k = \begin{pmatrix} A_k & B_k \\ B_k & 0 \end{pmatrix},$$

where the matrix A_k corresponds to the bilinear form a of Theorem 1, and the matrix B_k describes the mortar finite element coupling corresponding to the bilinear form \hat{d} . Starting with a vector z_k^0 , the multigrid iteration for the solution of (11) consists of preconditioning steps

$$z_k^{i+1} = z_k^i + M_k(d_k^i) \quad \text{with} \quad d_k^i = f_k - K_k z_k^i,$$

where the multigrid corrections $c_k^i = M_k(d_k^i)$ are defined recursively. On the coarser levels, $l = 0, \dots, k-1$, we need the stiffness matrix K_l and the restricted defect d_l . Appropriate grid transfer and smoothing matrices are required:

The prolongation matrix P_l corresponding to the grid transfer $V_{l-1} \rightarrow V_l$ can be constructed as a block diagonal matrix for the three ansatz spaces. On $S_1(\Omega_2, \mathcal{T}_{h_2})$ we choose piecewise linear interpolation. In case of problem 2, we replace the piecewise linear elements on Ω_2 by piecewise bilinear elements, and use the bilinear interpolation operator. On $CR(\Omega_1, \mathcal{T}_{h_1})$, we use the averaged interpolation introduced by Brenner [18], and for $M(\Gamma; \mathcal{E}_{h_1})$ a piecewise constant interpolation. The restriction matrix is the transposed operator $R_l = P_l^T$.

For the smoothing process of the correction vector

$$(12) \quad c_l^{m+1} = c_l^m + \tilde{K}_l^{-1}(d_l - K_l c_l^m),$$

we consider a smoothing matrix of the following form

$$\tilde{K}_l := \begin{pmatrix} \tilde{A}_l & B_l^T \\ B_l & 0 \end{pmatrix} = \begin{pmatrix} \tilde{A}_l & 0 \\ B_l & -\tilde{S}_l \end{pmatrix} \begin{pmatrix} 1 & \tilde{A}_l^{-1} B_l^T \\ 0 & 1 \end{pmatrix},$$

where $\tilde{A}_l := (\text{diag}(A_l) - \text{lower}(A_l))\text{diag}(A_l)^{-1}(\text{diag}(A_l) - \text{upper}(A_l))$ is the symmetric Gauß-Seidel decomposition of A_l . The construction of the approximated Schur complement \tilde{S}_l is motivated by the following Lemma. In case of the Stokes problem, it is given in [16], (Lemma 3.2).

LEMMA 3. If $\tilde{S}_l := B_l \tilde{A}_l^{-1} B_l^T$, then the iteration (12) satisfies the smoothing property

$$\|K_l(c_l - c_l^m)\| \leq c \frac{\|\tilde{A}_l\|}{m} \|c_l\|$$

where c_l^0 is the start iterate and $c_l := K_l^{-1}d_l$ denotes the exact correction for the given defect d_l on level l .

TABLE 1. Asymptotic convergence rates (average over a defect reduction of 10^{-10})

Example 1			Example 2		
number of elements		conv. rate	number of elements		conv. rate
Ω_1	Ω_2		Ω_1	Ω_2	
2048	32	0.24	1024	1024	0.22
8192	128	0.23	4096	4096	0.22
32768	512	0.19	16384	16384	0.21
131072	2048	0.21	65536	65536	0.19

For H^2 -regular domains, a L^2 -estimate can be derived and similar to [17] the approximation property can be obtained. This proves W-cycle convergence for the multigrid method. Note, that due to the nonconforming part in the discretization the Galerkin property does not hold ($K_{l-1} \neq R_l K_l P_l$) and standard methods for the proof of V-cycle convergence fail.

For our numerical experiments, the Schur complement $B_l \tilde{A}_l^{-1} B_l^T$ of the smoother \tilde{K}_l is replaced by the damped symmetric Gauß-Seidel decomposition of the approximate Schur complement

$$\hat{S}_l = B_l \operatorname{diag}(A_l)^{-1} B_l^T.$$

Now, we can define the multigrid V-cycle $c_l = M_k(d_l)$.

For $l = 0$,

$$\text{set } c_0 = K_0^{-1} d_0.$$

For $l > 0$,

$$\text{set } c_l^0 = 0,$$

define $c_l^1, \dots, c_l^{\nu_0}$ by the smoothing iteration (12),

restrict the defect $d_{l-1} = R_l(d_l - K_l c_l^{\nu_0})$,

apply the coarse grid correction $c_l^{\nu_0+1} = c_l^{\nu_0} + P_l M_{l-1}(d_{l-1})$,

and post smoothing steps $c_l^{\nu_0+1}, \dots, c_l^{\nu_0+1+\nu_1}$;

$$\text{set } c_l = c_l^{\nu_0+1+\nu_1}.$$

In the computations, we use $\nu_0 = \nu_1 = 2$.

The convergence rates are given in Table 1. In both cases robust results and asymptotic convergence rates independent of the refinement level are obtained. Multigrid V-cycles with transforming smoothers are efficient iterative solvers for saddle point problems arising from mortar finite element discretizations.

References

1. G. Abdoulaev, Y. Kuznetsov, and O. Pironneau, *The numerical implementation of the domain decomposition method with mortar finite elements for a 3D problem*, Tech. report, Laboratoire d'Analyse Numérique, Univ. Pierre et Marie Curie, Paris, 1996.
2. Y. Achdou and Y. Kuznetsov, *Substructuring preconditioners for finite element methods on nonmatching grids*, East-West J. Numer. Math. **3** (1995), 1–28.
3. Y. Achdou, Y. Kuznetsov, and O. Pironneau, *Substructuring preconditioners for the Q_1 mortar element method*, Numer. Math. **71** (1995), 419–449.
4. Y. Achdou, Y. Maday, and O. Widlund, *Méthode itérative de sous-structuration pour les éléments avec joints*, C. R. Acad. Sci., Paris, Ser. I **322** (1996), 185–190.
5. ———, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, Tech. Report 735, Department of Computer Science, Courant Institute, 1997.

6. T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, *Mixed finite element methods on non-matching multiblock grids*, Tech. Report TICAM 96-50, Texas Inst. Comp. Appl. Math., University of Texas at Austin, 1996, Submitted to SIAM J. Num. Anal.
7. T. Arbogast and I. Yotov., *A non-mortar mixed finite element method for elliptic problems on non-matching multiblock grids*, Comput. Meth. Appl. Mech. Eng. (1997), To appear.
8. D.N. Arnold and F. Brezzi, *Mixed and nonconforming finite element methods: Implementation, post-processing and error estimates*, M^2AN Math. Model. Num. Anal. **19** (1985), 7–35.
9. P. Bastian, *Parallele adaptive Mehrgitterverfahren*, Teubner Skripten zur Numerik, Teubner-Verlag, 1996.
10. P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuß, H. Rentz-Reichert, and C. Wieners, *UG – a flexible software toolbox for solving partial differential equations*, Computing and Visualization in Science **1** (1997), 27–40.
11. F. Ben Belgacem, *The mortar finite element method with Lagrange multipliers*, Tech. report, Laboratoire d'Analyse Numérique, Univ. Pierre et Marie Curie, Paris, 1995, To appear in Numer. Math.
12. F. Ben Belgacem and Y. Maday, *The mortar element method for three dimensional finite elements*, M^2AN **31** (1997), 289–302.
13. C. Bernardi and Y. Maday, *Raffinement de maillage en éléments finis par la méthode des joints*, C. R. Acad. Sci., Paris, Ser. I **320** (1995), 373–377.
14. C. Bernardi, Y. Maday, and A.T. Patera, *Domain decomposition by the mortar element method*, In: Asymptotic and numerical methods for partial differential equations with critical parameters (H. Kaper et al., ed.), Reidel, Dordrecht, 1993, pp. 269–286.
15. ———, *A new nonconforming approach to domain decomposition: the mortar element method*, In: Nonlinear partial differential equations and their applications (H. Brezzi et al., ed.), Paris, 1994, pp. 13–51.
16. D. Braess and R. Sarazin, *An efficient smoother for the Stokes problem*, Applied Numer. Math. **23** (1997), 3–19.
17. D. Braess and R. Verfürth, *Multigrid methods for nonconforming finite element methods*, SIAM Num. Anal. **27** (1990), 979–986.
18. S. C. Brenner, *An optimal order multigrid method for P1 nonconforming finite elements*, Math. Comp. **52** (1989), 1–15.
19. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag, New York, 1991.
20. M. Casarin and O. Widlund, *A hierarchical preconditioner for the mortar finite element method*, ETNA **4** (1996), 75–88.
21. Y. Kuznetsov and M. F. Wheeler, *Optimal order substructuring preconditioners for mixed finite elements on non-matching grids*, East-West J. Numer. Math. **3** (1995), 127–143.
22. J. Pousin and T. Sassi, *Adaptive finite element and domain decomposition with non matching grids*, 2nd ECCOMAS Conf. on Numer. Meth. in Engrg., Paris (J.-A. Désidéri et al., ed.), Wiley, Chichester, 1996, pp. 476–481.
23. G. Wittum, *On the convergence of multigrid methods with transforming smoothers. Theory with application to the Navier-Stokes equations*, Numer. Math. **54** (1989), 543–563.
24. B. Wohlmuth, *Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers*, Tech. Report 749, Courant Institute of Math. Sciences, New York University, 1997, Submitted.
25. ———, *A residual based error estimator for mortar finite element discretizations*, Tech. Report 370, Math. Inst., University of Augsburg, 1997, Submitted.
26. I. Yotov, *A mixed finite element discretization on non-matching multiblock grids for a degenerate parabolic equation arising in porous media flow*, East-West J. Numer. Math. **5** (1997), 211–230.

INST. FÜR COMPUTERANWENDUNGEN III, UNIVERSITÄT STUTTGART, PFAFFENWALDRING 27,
D-70550 STUTTGART, GERMANY.

E-mail address: wieners@ica3.uni-stuttgart.de

MATH. INSTITUTE, UNIVERSITY OF AUGSBURG, D-86135 AUGSBURG, GERMANY.

Current address: Courant Institute of Mathematical Sciences, 251 Mercer Street, New York,
N.Y. 10012, USA.

E-mail address: wohlmuth@cs.nyu.edu, wohlmuth@math.uni-augsburg.de