Copper Mountain Conf. on Iterative Methods

April 9-13, 1996

Copper Mountain, Colorado

Proceedings supported by

Cray Research, Inc.

Volume II

Organized by

Front Range Scientific Computations, Inc.

The University of Colorado

Sponsored by

Department of Energy
National Science Foundation

In cooperation with

SIAM Special Interest Group on
Numerical Linear Algebra

CONFERENCE CHAIRMEN

Tom Manteuffel
Steve McCormick

PROGRAM COMMITTEE

Loyce Adams
Steve Ashby
Howard Elman
Roland Freund
Anne Greenbaum
Seymour Parter
Paul Saylor
Nick Trefethen
Hank van der Vorst
Homer Walker
Olof Widlund

DISCLAIMER

TUESDAY, APRIL 9TH

SESSION I

Topic: *Nonlinear*	*Session Chair:* *Homer Walker*	*Room A*
8:00 - 8:30	M.D. Tocci	Method of Lines Solution of Richards' Equation
8:30 - 9:00	C.T. Kelley	A Multilevel Method for Conductive-Radiative Heat Transfer
9:00 - 9:30	H. Walker	An Adaption of Krylov Subspace Methods to Path Following
9:30 - 10:00	J. Neuberger	A Numerical Method for Finding Sign-Changing Solutions of Superlinear Dirichlet Problems

Topic: *Parallel*	*Session Chair:* *Loyce Adams*	*Room B*
8:00 - 8:30	A. Basermann	Parallel Preconditioning Techniques for Sparse CG Solvers
8:30 - 9:00	M. Field	Optimising a Parallel Conjugate Gradient Solver
9:00 - 9:30	A. Grama	Parallel Iterative Solvers and Preconditioners Using Approximate Hierarchical Methods
9:30 - 10:00	G. Li	A Block Variant of the GMRES Method on Massively Parallel Processors

Topic: *Preconditioning*	*Session Chair:* *Seymour Parter*	*Room C*
8:00 - 8:30	C. Brooking	Using Sparse LU Factorisation to Precondition GMRES for a Family of Similarly Structured Matrices Arising from Process Modelling
8:30 - 9:00	C.H. Guo	Incomplete Block Factorization Preconditioning for Indefinite Elliptic Problems
9:00 - 9:30	S.D. Kim	Preconditioning Cubic Spline Collocation Method by FEM and FDM for Elliptic Equations
9:30 - 10:00	S. Parter	Preconditioning Chebyshev Spectral Methods by Finite-Element and Finite-Difference Methods

SESSION II

Topic: *Nonlinear*	*Session Chair:* *Homer Walker*	*Room A*
10:30 - 11:00	D. Knoll	Enhanced Nonlinear Iterative Techniques Applied to a Non-Equilibrium Plasma Flow
11:00 - 11:30	V. Pan	Newton's Iteration for Inversion of Cauchy-like and Other Structured Matrices
11:30 - 12:00	M. Drexler	Fractal Aspects and Convergence of Newton's Method

Topic: *Parallel*	*Session Chair:* *Loyce Adams*	*Room B*
10:30 - 11:00	G.C. Lo	Iterative Solution of General Sparse Linear Systems on Clusters of Workstations
11:00 - 11:30	Q. Yao	New Concurrent Iterative Methods with Monotonic Convergence
11:30 - 12:00	R. McLay	Improving Matrix-Vector Product Performance and Multi-Level Preconditioning for the Parallel PCG Package

Topic: *Preconditioning*	*Session Chair:* *Seymour Parter*	*Room C*
10:30 - 11:00	M. Tuma	Approximate Inverse Preconditioning of Iterative Methods for Nonsymmetric Linear Systems
11:00 - 11:30	I. Mishev	Comparison of Different Preconditioners for Nonsymmetric Finite Volume Element Methods
11:30 - 12:00	C.W. Oosterlee	An Evaluation of Parallel Multigrid as a Solver and a Preconditioner for Singular Perturbed Problems

SESSION III

Topic: *Nonlinear*	*Session Chair:* *Homer Walker*	*Room A*
4:45 - 5:15	M. Pernice	NITSOL: A Newton Iterative Solver for Nonlinear Systems
5:15 - 5:45	M. Trummer	Nonsymmetric Systems Arising in the Computation of Invariant Tori
5:45 - 6:15	R. Renaut	Multisplitting for Linear, Least Squares and Nonlinear Problems

Topic: *Parallel*	*Session Chair:* *Loyce Adams*	*Room B*
4:45 - 5:15	V. Menkov	Solving Block Linear Systems with Low-Rank Off-Diagonal Blocks is Easily Parallelizable
5:15 - 5:45	Y. Shapira	Parallelizable Approximate Solvers for Recursions Arising in Preconditioning
5:45 - 6:15	D. Xie	New Parallel SOR Method by Domain Partitioning

Topic: *Preconditioning*	*Session Chair:* *Seymour Parter*	*Room C*
4:45 - 5:15	J.D. Moulton	Approximate Schur Complement Preconditioning of the Lowest Order Nodal Discretizations
5:15 - 5:45	K. Chen	On Preconditioning Techniques for Dense Linear Systems Arising from Singular Boundary Integral Equations
5:45 - 6:15	W.L. Wan	Fast Wavelet Based Sparse Approximate Inverse Preconditioner
6:15 - 6:45	E. Meese	Combined Incomplete LU and Strongly Implicit Procedure Preconditioning

Room TBA	*Workshop Chair*	
7:30 p.m.	*I. Duff*	Sparse Matrix Test Collections

THURSDAY, APRIL 11TH

SESSION I

Topic: *Navier-Stokes*	*Session Chair:* *Howard Elman*	*Room A*
8:00 - 8:30	V. Sarin	An Efficient Iterative Method for the Generalized Stokes Problem
8:30 - 9:00	P. Fischer	A Deflation based Parallel Algorithm for Spectral Element Solution of the Incompressible Navier-Stokes Equations
9:00 - 9:30	A. Wathen	An Iteration for Indefinite and Non-Symmetric Systems and its Application to the Navier-Stokes Equations
9:30 - 10:00	H. Elman	Perturbation of Eigenvalues of Preconditioned Navier-Stokes Operators

Topic: *Krylov Methods*	*Session Chair:* *M. Gutknecht*	*Room B*
8:00 - 8:30	T. DeLillo	Numerical Conformal Mapping Methods for Exterior and Doubly Connected Regions
8:30 - 9:00	M. Sosonkina	A New Adaptive GMRES Algorithm for Achieving High Accuracy
9:00 - 9:30	E. de Sturler	Truncation Strategies for (Nested) Krylov Methods
9:30 - 10:00	K. Ressel	Hybrid Lanczos-Type Product Methods

Topic: *Eigenvalues*	*Session Chair:* *Homer Walker*	*Room C*
8:00 - 8:30	K. Wu	Preconditioned Krylov Subspace Methods for Eigenvalue Problems
8:30 - 9:00	J. Baglama	A Numerical Method for Eigenvalue Problems in Modeling Liquid Crystals
9:00 - 9:30	A. Knyazev	A Subspace Preconditioning Algorithm for Eigenvector/Eigenvalue Computation
9:30 - 10:00	Z. Drmac	Stable Computation of Generalized Singular Values

SESSION II

Topic: *Navier-Stokes*	*Session Chair:* *Howard Elman*	*Room A*
10:30 - 11:00	I. Yavneh	Fast Multigrid Solution of the Advection Problem with Closed Characteristics
11:00 - 11:30	A.A. Lorber	Accelerated Solution of Non-Linear Flow Problems Using Chebyshev Iteration Polynomial Based Runge-Kutta Recursions
11:30 - 12:00	M. Murphy	Towards an Ideal Preconditioner for Linearized Navier-Stokes Problems

Topic: *Krylov Methods*	*Session Chair:* *M. Gutknecht*	*Room B*
10:30 - 11:00	M. Gutknecht	Look-Ahead Procedures for Lanczos-Type Product Methods Based on Three-Term Recurrences
11:00 - 11:30	E. Gallopoulos	Solving Modified Systems with Multiple Right-Hand Sides
11:30 - 12:00		

Topic: *Eigenvalues*	*Session Chair:* *Homer Walker*	*Room C*
10:30 - 11:00	A. Stathopoulos	Thick Restarting of the Davidson Method: an Extension to Implicit Restarting
11:00 - 11:30	X. Zou	Splitting the Determinants of Upper Hessenberg Matrices & the Hyman Method
11:30 - 12:00	M. Fernandes	A Combined Modification of the Newton's Method for Systems of Nonlinear Equations

Topic: *Student Papers Winners*	*Session Chair:* *T. Manteuffel & S. McCormick*	*Room A*
4:45 - 5:15	S. Knapek	Matrix-Dependent Multigrid-Homogenization for Diffusion Problems
5:15 - 5:45	M. Horn	A Superlinear Convergence Estimate for an Iterative Method for the Biharmonic Equation
5:45 - 6:15	A. Klawonn	Triangular Preconditioners for Saddle Point Problems with a Penalty Term

Cash Bar *Room TBA*
6:45 - 7:30 p.m.

Banquet *Room TBA*
7:30 - 9:30 p.m.

FRIDAY, APRIL 12TH

SESSION I

Topic: *Domain Decomp.*	*Session Chair:* *Olof Widlund*	*Room A*
8:00 - 8:30	X.C. Cai	Newton-Krylov-Schwarz Algorithms for the 2D Full Potential Equations
8:30 - 9:00	D. Keyes	Newton-Krylov-Schwarz Methods in Unstructured Grid Euler Flow
9:00 - 9:30	U. Trottenberg	Adaptive Parallel Multigrid for Euler & Incompressible Navier-Stokes Equations
9:30 - 10:00	O. Widlund	Domain Decomposition Methods for Mortar Finite Elements
Topic: *Krylov Methods*	*Session Chair:* *Anne Greenbaum*	*Room B*
8:00 - 8:30	V. Druskin	Extended Krylov Subspaces Approximations of Matrix Functions; Application to Computational Electromagnetics
8:30 - 9:00	D. Sorensen	Krylov Subspace Methods for Computation of Matrix Functions
9:00 - 9:30	T. Tamarchenko	Application of Spectral Lanczos Decomposition Method to Large Scale Problems Arising Geophyics
9:30 - 10:00	A. Greenbaum	Some Uses of the Symmetric Lanczos Algorithm and Why it Works!
Topic: *CFD*	*Session Chair:* *TBA*	*Room C*
8:00 - 8:30	X. Zheng	Multigrid Solution of Incompressible Turbulent Flows by Using Two-Equation Turbulence Models
8:30 - 9:00	S. Kumar	Nonlinear Krylov Acceleration of Reacting Flow Codes
9:00 - 9:30	M.D. Tidriri	Schwarz-based Algorithms for Compressible Flows
9:30 - 10:00	C. Liao	Multilevel Local Refinement and Multigrid Methods for 3-D Turbulent Flow
Topic: *Doman Decomp.*	*Session Chair:* *Olof Widlund*	*Room A*
10:30 - 11:00	X. Feng	A Mixed Finite Element Domain Decomposition Method for Solving Nearly Elastic Wave Equations in the Frequency Domain
11:00 - 11:30	A. Jemcov	Representation of Discrete Steklov-Poincare Operator Arising in Domain Decomposition Methods in Wavelet Basis
11:30 - 12:00	J. Xu	Simplified Approach to Some Nonoverlapping Domain Decomp. Methods
Topic: *Krylov Methods*	*Session Chair:* *Anne Greenbaum*	*Room B*
10:30 - 11:00	F. Campos	The Adaptive CCCG(n) Method for Efficient Solution of Time Dependent Partial Differential Equations
11:00 - 11:30	T. Barth	Conjugate Gradient Algorithms Using Multiple Recursions
11:30 - 12:00	J. Cullum	Iterative Methods for Solving Ax=b, GMRES/FOM versus QMR/BiCG

SESSION III

Topic: *Domain Decomp.*	*Session Chair:* *Olof Widlund*	*Room A*
4:45 - 5:15	S. Maliassov	Domain Decomposition Method for Nonconforming Finite Element Approximations of Anisotropic Elliptic Problems on Nonmatching Grids
5:15 - 5:45	H. Zhao	Analysis of Generalized Schwarz Alternating Procedure for Domain Decomposition
5:45 - 6:15	R. Tezaur	Substructuring by Lagrange Multipliers for Solids and Plates
6:15 - 6:45	X.C. Tai	Some Nonlinear Space Decomposition Algorithms

Topic: *Krylov Methods*	*Session Chair:* *Anne Greenbaum*	*Room B*
4:45 - 5:15	E. Bobrovnikova	Iterative Methods for Weighted Least-Squares
5:15 - 5:45	A. Lumsdaine	Krylov Subspace Acceleration of Waveform Relaxation
5:45 - 6:15	C. Wagner	Tangential Frequency Filtering Decompositions
6:15 - 6:45	J. Zhang	Multigrid Solution of Convection-Diffusion Equation with High-Reynolds Number

Topic: *Markov Chains*	*Session Chair:* *Daniel Szyld*	*Room C*
4:45 - 5:15	T. Dayar	State Space Orderings for Gauss-Seidel in Markov Chains Revisited
5:15 - 5:45	G. Horton	On the Multi-Level Solution Algorithm for Markov Chains
5:45 - 6:15	D. Szyld	Threshold Partitioning of Sparse Matrices and Applications to Markov Chains
6:15 - 6:45		

Room TBA	*Workshop Chair*	
7:30 p.m.	*Mike Heroux*	Sparse and Parallel BLAS

SATURDAY, APRIL 12TH

SESSION I

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*	*Room A*
8:00 - 8:30	R. Alchalabi	Multigrid Method Applied to the Solution of an Elliptic, Generalized Eigenvalue Problem
8:30 - 9:00	J. Dendy	Some Multigrid Algorithms for SIMD Machines
9:00 - 9:30	C. Douglas	Multigrid on Unstructured Grids Using an Auxiliary Set of Structured Grids
9:30 - 10:00	M. Griebel	Multiscale Iterative Methods, Coarse Level Operator Construction and Discrete Homogenization Techniques

Topic: *Applications*	*Session Chair:* *TBA*	*Room B*
8:00 - 8:30	H.C. Chen	Embedding SAS Approach Into Conjugate Gradient Algorithms for Asymmetric 3D Elasticity Problems
8:30 - 9:00	M. Clemens	Iterative Methods for the Solution of Very Large Complex-Symmetric Linear Systems of Equations in Electrodynamics
9:00 - 9:30	T. Cwik	Matrix Equation Decomposition and Parallel Solution of Systems Resulting from Unstructured Finite Element Problems in Electromagnetics
9:30 - 10:00	S.W. Bova	Iterative Solution of the Semiconductor Device Equations

Topic: *Multiple RHS*	*Session Chair:* *Roland Freund*	*Room C*
8:00 - 8:30	W. Boyse	Multiple Solutions to Dense Systems in Radar Scattering using a Preconditioned Block GMRES Solver
8:30 - 9:00	R. Freund	The BL-QMR Algorithm for Non-Hermitian Linear Systems with Multiple Right-Hand Sides
9:00 - 9:30	M. Malhotra	Iterative Solution of Multiple Radiation and Scattering Problems in Structural Acoustics Using the BL-QMR Algorithm
9:30 - 10:00	T. Chan	Galerkin Projection Methods for Solving Multiple Related Linear Systems

SESSION II

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*	*Room A*
10:30 - 11:00	R. Hornung	Adaptive Mesh Refinement and Multilevel Iteration for Multiphase, Multicomponenet Flow in Porous Media
11:00 - 11:30	J. Jones	Semi-Coarsening Multigrid Methods for Parallel Computing
11:30 - 12:00	P. Vanek	An Algebraic Multigrid Algorithm for Symmetric Positive Definite Linear Systems

Topic: *Applications*	*Session Chair:* *TBA*		*Room B*
10:30 - 11:00	A. Frommer	Lattice QCD Computations: Recent Progress with Modern Krylov Subspace Methods	
11:00 - 11:30	R. Karamikhova	Numerical Solution of High-Kappa Model of Superconductivity	
11:30 - 12:00	M. Heroux	The Impact of Improved Sparse Linear Solvers on Industrial Engineering Applications	

Topic: *Projection Methods*	*Session Chair:* *Roland Freund*		*Room C*
10:30 - 11:00	A. Popov	Projection Preconditioning for Lanczos-Type Methods	
11:00 - 11:30	R. Bramley	Partial Row Projection Methods	
11:30 - 12:00	P. Kolm	Generalized Subspace Correction Methods	

SESSION III

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*		*Room A*
4:45 - 5:15	S. Oliveira	A Multigrid Method for Variational Inequalities	
5:15 - 5:45	W. Schmid	A Multigrid Solution Method for Mixed Hybrid Finite Elements	
5:45 - 6:15	H.J. Bungartz	A Unidirectional Approach for d-Dimensional Finite Element Methods of Higher Order on Sparse Grids	
6:15 - 6:45	M. Brezina	Two-Level Method with Coarse Space Size Independent Convergence	

Topic: *Applications*	*Session Chair:* *Tom Russell*		*Room B*
4:45 - 5:15	C. Yang	Numerical Computation of the Linear Stability of the Diffusion Model for Crystal Growth Simulation	
5:15 - 5:45	L. Borges	Highly Indefinite Multigrid for Eigenvalue Problems	
5:45 - 6:15	G.S. Lett	An Adaptive Nonlinear Solution Scheme for Reservoir Simulation	
6:15 - 6:45	A. Cardona	An Iterative Method to Invert the LTSn Matrix	

Topic: *Helmholtz*	*Session Chair:* *Roland Freund*		*Room C*
4:45 - 5:15	E. Larsson	Iterative Solution of the Holmholtz Equation	
5:15 - 5:45	S. Kim	Iterative Procedures for Wave Propagation in the Frequency Domain	
5:45 - 6:15	J. Yoo	Multigrid for the Galerkin Least Squares Method in Linear Elasticity: The Pure Displacement Problem	
6:15 - 6:45			

FRIDAY, APRIL 12TH

Topic: *Domain Decomp.*	*Session Chair:* *Olof Widlund*	*Room A*
8:00 - 8:30	X.C. Cai	Newton-Krylov-Schwarz Algorithms for the 2D Full Potential Equations
8:30 - 9:00	D. Keyes	Newton-Krylov-Schwarz Methods in Unstructured Grid Euler Flow
9:00 - 9:30	U. Trottenberg	Adaptive Parallel Multigrid for Euler & Incompressible Navier-Stokes Equations
9:30 - 10:00	O. Widlund	Domain Decomposition Methods for Mortar Finite Elements

NEWTON-KRYLOV-SCHWARZ ALGORITHMS FOR THE 2D FULL POTENTIAL EQUATION

XIAO-CHUAN CAI*, WILLIAM D. GROPP†, DAVID E. KEYES‡, ROBIN G. MELVIN§ AND DAVID P. YOUNG¶

Abstract. We study parallel two-level overlapping Schwarz algorithms for solving nonlinear finite element problems, in particular, for the full potential equation of aerodynamics discretized in two dimensions with bilinear elements. The main algorithm, Newton-Krylov-Schwarz (NKS), employs an inexact finite-difference Newton method and a Krylov space iterative method, with a two-level overlapping Schwarz method as a preconditioner. We demonstrate that NKS, combined with a density upwinding continuation strategy for problems with weak shocks, can be made robust for this class of mixed elliptic-hyperbolic nonlinear partial differential equations, with proper specification of several parameters. We study upwinding parameters, inner convergence tolerance, coarse grid density, subdomain overlap, and the level of fill-in in the incomplete factorization, and report favorable choices for numerical convergence rate and overall execution time on a distributed-memory parallel computer.

1. Introduction. In the past few years domain decomposition methods for linear partial differential equations, including overlapping Schwarz methods [9, 26], have graduated from theory into practice in many applications [19]. In this paper, we study several aspects of the parallel implementation of a Krylov-Schwarz domain decomposition algorithm for the finite element solution of the nonlinear full potential equation of aerodynamics, extending our model studies of linear convection-diffusion problems in [3] and of linear aerodynamic design optimization problems in [23]. Newton-Krylov methods [1, 10, 11, 28] are potentially well suited and increasingly popular for the implicit solution of nonlinear problems whenever it is expensive to compute or store a true Jacobian. We employ a combined algorithm, called Newton-Krylov-Schwarz, and focus on the interplay of the three nested components of the algorithm, since the amount of work done in each component affects and is affected by the work done in the others.

Newton-Krylov-Schwarz is a general purpose parallel solver for nonlinear partial differential equations and has been applied to complex multicomponent systems of compressible and reacting flows in, e.g. [5, 6, 20]. This paper is concerned with the simpler scalar problem of the full potential equation, which describes inviscid, irrotational, isentropic compressible flow. Though the full potential model is highly idealized, it remains the model of choice of external aerodynamic designers to date, because codes based thereupon offer reasonable turnaround times and in many cases high accuracy compared to state-of-the-art Navier-Stokes solvers. Though derived under the condition of isentropy, the full potential model

* Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309. *cai@cs.colorado.edu.* This work was supported in part by NSF grants ASC-9457534, ASC-9217394, and ECS-9527169, by NASA grant NAG5-2218, and by NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering.

† Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. *gropp@mcs.anl.gov.* This work was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

‡ Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 and ICASE, NASA Langley Research Center, Hampton, VA 23681. *keyes@icase.edu.* This work was supported in part by NSF grants ECS-8957475 and ECS-9527169, by the State of Connecticut and the United Technologies Research Center, and by NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering.

§ The Boeing Company, Seattle, WA 98124. *rgm4152@cfdd53.cfd.ca.boeing.com.*

¶ The Boeing Company, Seattle, WA 98124. *dpy6629@cfdd51.cfd.ca.boeing.com.*

remains useful in flows with weak shocks, with pre-shock Mach numbers of about 1.2 or less. It can also be extended by boundary layer patching to incorporate viscous effects, by a branch cut to accommodate lift, and by source terms to simulate powered engines. In engineering practice, accurately modeling such nonideal effects in complex geometries accounts for almost all of the lines of code, but the solution of the resulting discrete equations accounts for the majority of the execution time. The lower per-cell storage and computational requirements of the potential model allow the use of grids dense enough to achieve low truncation error levels for complex geometries. The full potential equation also avoids the spurious entropy generation near stagnation often associated with Euler and Navier-Stokes codes for industrial complex geometries of interest. We justify the simply coded examples in this paper by our focus on a solution algorithm that should not require any changes other than greater irregularity in its sparse data structures to be useful in more practical settings.

With Newton's method as the outer iteration, a highly nonsymmetric and/or indefinite large, sparse Jacobian equation needs to be solved at every iteration to a certain accuracy, which is often progressively tightened in response to a falling nonlinear residual norm. The most popular family of preconditioners for large sparse Jacobians on structured or unstructured grids, incomplete factorization, is difficult to parallelize efficiently flop-for-flop in its original global form. In our approach, the ILU-preconditioner for the Newton correction equations is replaced by a multi-level overlapping Schwarz preconditioner. The latter is not only scalably parallelizable up to available parallel granularities, but also possesses an asymptotically optimal mesh- and granularity-independent convergence rate for elliptically dominated problems. Our two-level overlapping additive Schwarz algorithm uses a non-nested coarse space. Subdomain granularity, quality of subdomain solves, coarse grid density, strategy for coarse grid solution, and inner iteration termination criteria are important factors in overall performance. We report numerical experiments on an IBM SP2 with up to 32 processors.

This report is a short version of [4].

2. The full potential problem. We study the full potential equation of aerodynamics, see [16],

$$\nabla \cdot (\rho v) = 0, \tag{1}$$

where $v = (v_1, v_2)^T$ is the velocity and ρ is the local density, respectively. We assume that the flow is irrotational, which implies that there exists a velocity potential Φ such that $v = \nabla\Phi$. Here

$$\rho(\Phi) = \rho_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{||\nabla\Phi||_2^2}{q_\infty^2})\right)^{1/(\gamma-1)}. \tag{2}$$

Observe that while the density is positive in regions of validity, (1) may be locally hyperbolic. We consider only subsonic farfield boundaries. Following Boeing's TRANAIR code [30], we employ a finite element formulation of the two-dimensional full potential equation using bilinear elements. The existence, uniqueness, and regularity of the solution are not central to this paper, but have been discussed in the papers [22, 24] and references therein. The finite element problem is formulated in terms of the weak form

$$a(\Phi, v) = \int_\Omega \rho(\Phi) \nabla\Phi \cdot \nabla v \, d\Omega.$$

For subsonic problems, the above mentioned finite element method is sufficient; however, for transonic cases upwinding has to be introduced in the density calculation in order to capture the weak shock in the solution. The proper use of an upwinding scheme is essential both to the success of the overall approach in finding the correct location and strength of the shock and to the convergence, or the fast convergence, of the inexact Newton's method.

Density ρ is assumed to be a constant in each element, and this constant is ordinarily determined by the four values of Φ at the corners of the element, through (2). Following [17, 30], if an element is determined to be supersonic, or nearly so, its density value is replaced by $\tilde{\rho} = \rho - \mu V \cdot \nabla_- \rho$, where V is the normalized element velocity and $\nabla_- \rho$ is an upwind undivided difference. Here μ is the element switching function,

$$\mu = \nu_0 \max\{0, 1 - M_c^2/M^2\}, \tag{3}$$

where M is the element Mach number, M_c is a pre-selected cutoff Mach number chosen to introduce dissipation just below Mach 1.0, and ν_0 is a constant usually set to something between 1.0 and 3.0 to increase the amount dissipation in the supersonic elements. In our implementation, we use a technique referred to as iterated maximization of the switching function. The details can be found in the our paper [4].

3. Newton-Krylov-Schwarz algorithms. NKS is a family of general purpose algorithms for solving nonlinear boundary value problems of partial differential equations. In terms of software development, NKS has three components that can be handled independently. However, to achieve reasonable overall convergence, the three components have to be tuned simultaneously.

Starting from an initial guess Φ_0, which is sufficiently close to the solution, a solution of the nonlinear system is sought by using an inexact Newton method: For some $\eta_k \in [0, 1)$ find s_k that satisfies

$$\|F(\Phi_k) + J(\Phi_k)s_k\| \leq \eta_k \tag{4}$$

and set $\Phi_{k+1} = \Phi_k + \lambda_k s_k$, where $\lambda_k \in (0, 1)$ is determined by a line search procedure [8]. The vector s_k is obtained by approximately solving the linear Jacobian system $J(\Phi_k)s_k = -F(\Phi_k)$ with a Krylov space iterative method. The action of Jacobian J on an arbitrary Krylov vector w can be approximated by $J(\Phi_k)w \approx \frac{1}{\epsilon}\left(F(\Phi_k + \epsilon w) - F(\Phi_k)\right)$. Finite-differencing with ϵ makes such matrix-free methods potentially more susceptible to finite word-length effects than ordinary Krylov methods. The most expensive component of the algorithm is the solution of the linear system with the Jacobian at each Newton iteration. As discussed in Eisenstat and Walker [11], when Φ_k is far from the solution, the local linear model used in deriving the Newton method may disagree considerably with the nonlinear function itself, and it is unproductive to "over-solve" these linear systems. We tested several stopping conditions, including those discussed in [11], and found that the best choice for our problems, based on elapsed execution time for a fixed relative nonlinear residual norm reduction, is simply to set $\eta_k = 10^{-2}\|F(\Phi_k)\|_2$.

We use the GMRES method [25], to solve the linear system of algebraic equations: $Px = b$, where P is the preconditioned Jacobian matrix. To fit the available memory, one is sometimes forced to use the k-step restarted GMRES method [25]. However, in this case neither an optimal convergence property nor even convergence is guaranteed. In our experiments, we do not need to solve the linear systems very accurately; i.e., $\eta = 10^{-2}$ in $\|b - Px_m\|_a \leq \eta\|r_0\|_2$ is sufficient to capture an accurate solution to the nonlinear problem,

in both subsonic and transonic cases. We do observe that, for certain maximum Krylov subspace dimensions (for example 30, in a problem with approximately 10^4 times as many discrete unknowns) and certain Mach numbers ($M_\infty = 0.8$), the restarted GMRES can never reduce the initial residual below 10^{-5}. In other words, there is no linear convergence. It is further noticed in such cases that the residual norm measured as a by-product in GMRES is no longer the same as, or even close to, the true residual norm except at the restarting points, where it is freshly updated.[1] A loose linear convergence tolerance avoids this problem by returning to the Newton method with a step that is far from exact. In the delicate balance between few nearly exact Newton steps with expensive inner linear solutions and many inexact Newton steps with bounded-cost inner linear solutions, we find the bottom line of overall execution time best served by bounding the inner linear work. This approach is also found most effective in the context of inviscid aerodynamics based on the primitive variable Euler equations in [6]. It deprives Newton's method of its asymptotic quadratic convergence, but provides steep linear convergence.

We use a two-level overlapping Schwarz preconditioner with inexact subdomain solvers and non-nested coarse grid to the linear system at each nonlinear iteration. Details can be found in [2].

4. Numerical results. In this section, we report some numerical results obtained on the IBM SP2 with up to 32 processors for both subsonic and transonic flows. Ω is a unit-aspect ratio square partitioned into a uniform rectangular meshes up to 512×512 in size. Let q_∞, the farfield flow speed, be normalized to 1. Let $\Phi_\infty = \int_x q_\infty dx$. On the farfield boundaries we assume $\Phi = \Phi_\infty$. On the symmetry boundary of Ω, the airfoil is located in $[1/3, 2/3]$ and we use the transpiration condition $\frac{\partial \Phi}{\partial y} = -\nabla \Phi_\infty \cdot (n_x, n_y)$, where $n = (n_x, n_y)$ is the unit outward normal, and where $y = f(x)$ describes the shape of airfoil. Once the function $f(x)$ is given, this condition becomes $\frac{\partial \Phi}{\partial y} = -q_\infty f'(x)$. On $[0, 1/3]$ and $[2/3, 1]$, we impose for symmetry the no penetration condition $\frac{\partial \Phi}{\partial n} = \frac{\partial \Phi}{\partial y} = 0$.

4.1. Observations — subsonic case. The linear systems that arise in this case fall within the elliptic theory for Schwarz [26]. It takes 6 Newton iterations to reduce the initial nonlinear residual by a factor of 10^{-10}. Because of the Krylov dimension cut-off, the convergence is linear; see the left panel in Fig. 1. Key observations from this example are as follows: (1) Even a modest coarse grid makes a significant improvement in an additive Schwarz preconditioner, especially when the number of subdomains is large. As much as 40% of the execution time can be saved when adding a 2×3 coarse grid to a no coarse grid preconditioner, for the 32-subdomain case. (2) A law of diminishing returns sets in at roughly one point per subdomain. (3) When using 8 processors, the total communication time is always less than 5% of the total computational time, however, it becomes as much as 26% when using 32 processors.

We also run results when the subproblems are solved with ILU(k) for various levels of fill-in. The overlap size is $3h$, and the coarse grid is 7×8. The conclusion from the tests is that the larger the k, the faster the method becomes. When using a small number of processors, like 8, the best execution time is obtained with ILU(5). However, if the processor number is large, the optimal result can only be obtained by considering several

[1] We believe, after Saad (personal communication), that this may be due to a lack of floating point commutativity in the product that expresses z_m in GMRES, namely $z_m = PV_m y$, where V_m is a Gram-Schmidt basis for K_m and y is a coefficient vector of dimension m that satisfies a related least squares problem (see [25]). The effect seems related to drastic variations in the magnitude of successive elements of y.

parameters: *ovlp*, k, the coarse mesh size, and perhaps others. We have not simultaneously varied all relevant parameters to get the best results, but have presented controlled slices through parameter space for insight.

4.2. Observations — transonic case. The first, and probably the most important, observation is that without a proper upwinding discretization, all three components of NKS can fail.

Fig. 3 shows the convergence history in terms of the C_p curves. We note that it takes only 4 to 5 iterations for the Newton's method to establish the neighborhood of the shock, but another 15 or so iterations to move it to the exact location. Mach contours at the final solution are given in Fig. 3. While the shock is setting up, the linear convergence of Newton's method is stalled; see the left panel of Fig. 2. The inclusion of a small coarse grid can reduce the total number of the linear iterations, as well as the total execution time, by a factor of 30%. An optimally chosen coarse grid size can lead to a greater savings. In Fig. 4, we overlay the convergence histories of all the linear solutions in a complete nonlinear calculation. The history in the left panel is without a coarse grid, and that in the right with a 7×8 coarse grid. The number of linear iterations and the total execution time can be reduced even further if a proper overlap size, which is not usually very small, is used.

The best result, in terms of the total execution time, among all the test calculations is obtained using a ILU(k), with $k = 5$, as the subproblems solver. It takes less than $2\frac{1}{2}$ minutes on the 32-processor IBM SP2 to set up and solve the Mach 0.8 nonlinear system with more than a quarter of a million unknowns.

5. Concluding remarks. We have investigated computationally the effectiveness of Newton-Krylov-Schwarz methods applied to the full potential equation of aerodynamics in some simplified situations in two space dimensions. Best performance is obtained with modest overlap, a modest coarse grid (one or two points per processor), modest-to-generous fill in the subdomain ILU preconditioners, and uniformly loose convergence tolerances on the Krylov iterations within each Newton step.

REFERENCES

[1] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton-Krylov algorithms*, SIAM J. Optimization, 4 (1994), pp. 297–330.

[2] X.-C. Cai, *The use of pointwise interpolation in domain decomposition methods with non-nested meshes*, SIAM J. Sci. Comput., 16 (1995), pp. 250–256.

[3] X.-C. Cai, W. D. Gropp, and D. E. Keyes, *A comparison of some domain decomposition and ILU preconditioned iterative methods for nonsymmetric elliptic problems*, Numer. Lin. Alg. Applics, 1 (1994), pp. 477–504.

[4] 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*, ICASE Report, 1996. (To appear)

[5] X.-C. Cai, W. D. Gropp, D. E. Keyes, and M. D. Tidriri, *Newton-Krylov-Schwarz methods in CFD*, in *Proceedings of the International Workshop on Numerical Methods for the Navier-Stokes Equations*, F. Hebeker and R. Rannacher, eds., Notes on Numerical Fluid Mechanics, Vieweg Verlag, Braunschweig (1994).

[6] X.-C. Cai, D. E. Keyes and V. Venkatakrishnan, *Newton-Krylov-Schwarz: An implicit solver for CFD*, in "Proc. of the Eighth International Conference on Domain Decomposition Methods in Science and Engineering" (R. Glowinski et al., eds.), Wiley, New York, 1996 (to appear).

[7] X.-C. Cai and O. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 243–258.

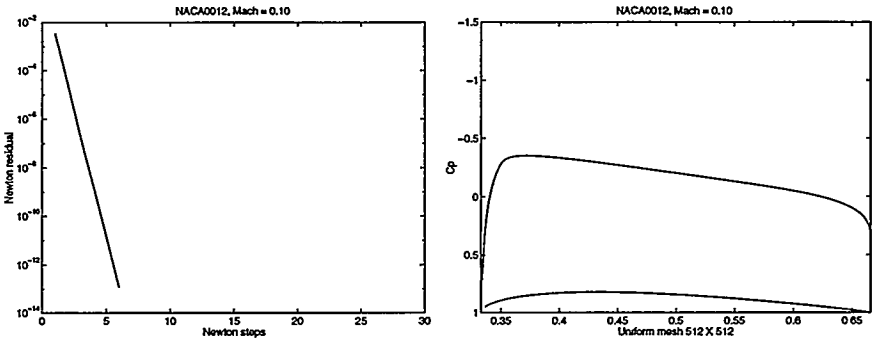


FIG. 1. $M_\infty = 0.1$. *History of the Newton residual, and C_p curve at convergence.*

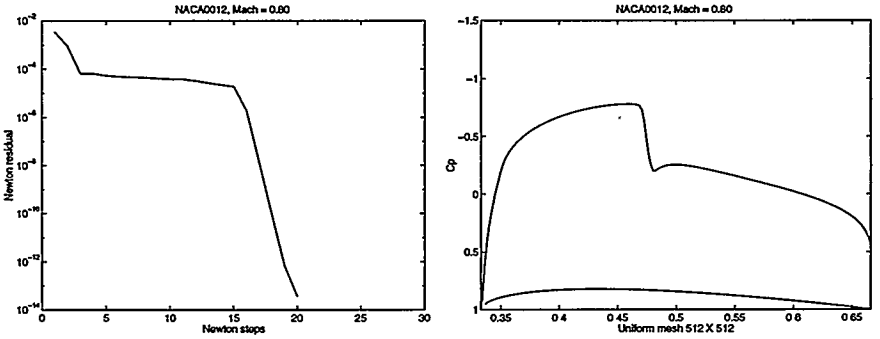


FIG. 2. $M_\infty = 0.8$. *History of the Newton residual, and C_p curve at convergence.*

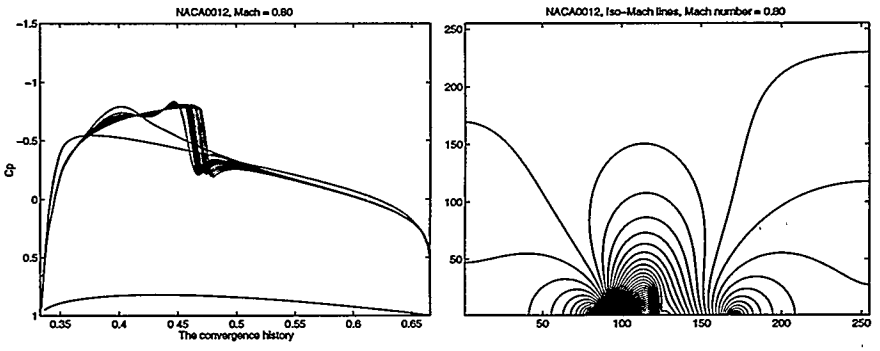


FIG. 3. $M_\infty = 0.8$. *History of C_p and Mach contours at convergence.*

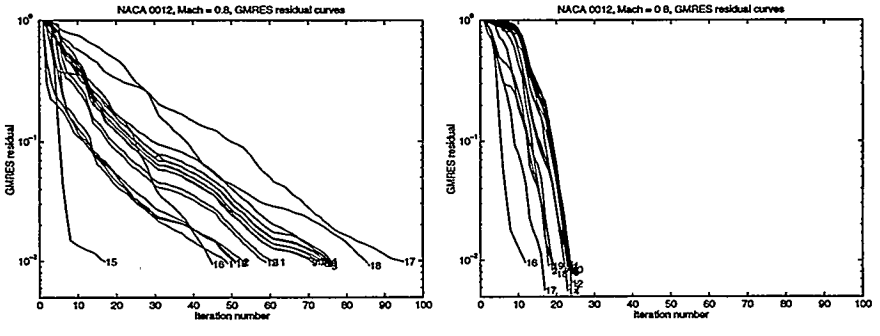


FIG. 4. *The GMRES convergence history with and without a coarse space*

[8] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, NJ, 1983.

[9] M. DRYJA AND O. B. WIDLUND, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, held in Houston, Texas, March 20-22, 1989, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1990.

[10] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.

[11] S. C. EISENSTAT AND H. F. WALKER, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[12] W. D. GROPP AND D. E. KEYES, *Domain decomposition on parallel computers*, Impact of Comp. in Sci. and Eng., 1 (1989), pp. 421–439.

[13] W. D. GROPP AND B. F. SMITH, *Users Manual for the Chameleon Parallel Programming Tools*, ANL-93/23, Argonne National Laboratory, 1993.

[14] ——, *Simplified Linear Equation Solvers Manual*, ANL-93/8, Argonne National Laboratory, 1993.

[15] ——, *Users Manual for KSP: Data-Structure-Neutral Codes Implementing Krylov Space Methods*, ANL-93/30, Argonne National Laboratory, 1993.

[16] C. HIRSCH, *Numerical Computation of Internal and External Flows*, 2 vols., Wiley, New York, 1990.

[17] T. L. HOLST AND W. F. BALLHAUS, *Fast, conservative schemes for the full potential equation applied to transonic flows*, AIAA J. 17 (1979), pp. 145–152.

[18] W. P. HUFFMAN, R. G. MELVIN, D. P. YOUNG, F. T. JOHNSON, J. E. BUSSOLETTI, M. B. BIETERMAN, AND C. L. HILMES, *Practical design and optimization in computational fluid dynamics*, AIAA Paper 93-3111, July 1993.

[19] D. E. KEYES, Y. SAAD AND D. G. TRUHLAR, eds., *Domain-based Parallel and Problem Decomposition Methods in Science and Engineering*, SIAM, Philadelphia, 1995.

[20] D. A. KNOLL, P. R. MCHUGH AND D. E. KEYES, *Newton-Krylov methods for low Mach number combustion*, in "Proc. of the 12th AIAA Computational Fluid Dynamics Conference" (San Diego, June 1995), AIAA Paper 95-1672 (accepted for AIAA Journal).

[21] C. LIU AND S. F. MCCORMICK, *Multigrid, elliptic grid generation and the fast adaptive composite grid method for solving transonic potential flow equations*, in Multigrid Methods: Theory, Applications, and Supercomputing, S. F. McCormick, ed., Lecture Notes in Pure and Appl. Math., 110, Marcel Dekker, New York, 1988, pp. 365-387.

[22] J. MANDEL AND J. NEČAS, *Convergence of finite elements for transonic potential flows*, SIAM J. Numer. Anal., 24 (1987), pp. 985–997.

[23] R. G. MELVIN, D. P. YOUNG, D. E. KEYES, C. C. ASHCRAFT, M. B. BIETERMAN, C. L. HILMES, W. P. HUFFMAN, AND F. T. JOHNSON, *A two-level iterative method applied to aerodynamic sensitivity calculations*, BCSTECH-94-047, Boeing Computer Services, December 1994.

[24] R. RANNACHER, *On the convergence of the Newton-Raphson method for strongly nonlinear elliptic problems*, in Nonlinear Computational Mechanics, P. Wriggers and W. Wagner eds., Springer-Verlag, 1991.

[25] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 865–869.

[26] B. F. SMITH, P. E. BJØRSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[27] S. TA'ASAN, G. KURUVILA, AND M. D. SALAS, *Aerodynamic Design and Optimization in One Shot*, AIAA Paper 92-0025.

[28] L. B. WIGTON, N. J. YU, AND D. P. YOUNG, *GMRES acceleration of computational fluid dynamics codes*, AIAA Paper 85-1494.

[29] D. P. YOUNG, R. G. MELVIN, M. B. BIETERMAN, F. T. JOHNSON, AND S. S. SAMANT, *Global convergence of inexact Newton methods for transonic flow*, International Journal for Numerical Methods in Fluids, 11 (1990), pp. 1075–1095.

[30] D. P. YOUNG, R. G. MELVIN, M. B. BIETERMAN, F. T. JOHNSON, S. S. SAMANT, AND J. E. BUSSOLETTI, *A locally refined rectangular grid finite element methods: Application to computational fluid dynamics and computational physics*, J. Comput. Phys., 92 (1991), pp. 1–66.

NEWTON-KRYLOV-SCHWARZ METHODS IN UNSTRUCTURED GRID EULER FLOW

DAVID E. KEYES*

Abstract. Newton-Krylov methods and Krylov-Schwarz (domain decomposition) methods have begun to become established in computational fluid dynamics (CFD) over the past decade. The former employ a Krylov method inside of Newton's method in a Jacobian-free manner, through directional differencing. The latter employ an overlapping Schwarz domain decomposition to derive a preconditioner for the Krylov accelerator that relies primarily on local information, for data-parallel concurrency. They may be composed as Newton-Krylov-Schwarz (NKS) methods, which seem particularly well suited for solving nonlinear elliptic systems in high-latency, distributed-memory environments. We give a brief description of this family of algorithms, with an emphasis on domain decomposition iterative aspects. We then describe numerical simulations with Newton-Krylov-Schwarz methods on an aerodynamic application emphasizing comparisons with a standard defect-correction approach and subdomain preconditioner consistency.

Key words. Euler equations, domain decomposition, Newton methods, Krylov space methods, overlapping Schwarz preconditioner, parallel computing.

AMS(MOS) subject classifications. 65H20, 65N55, 65Y05, 76G25.

1. Introduction. Several trends contribute to the importance of parallel implicit algorithms in CFD. Multidisciplinary analysis and optimization put a premium on the ability of algorithms to achieve low residual solutions rapidly, since analysis codes for individual components are typically solved iteratively and their results are often differenced for sensitivities. Problems possessing multiple scales provide the classical motivation for implicit algorithms and arise frequently in locally adaptive contexts or in dynamical contexts with multiple time scales, such as aero-elasticity. Meanwhile, the never slackening demand for resolution and prompt turnaround forces consideration of parallelism, and, for cost effectiveness, particularly parallelism of the high-latency, low-bandwidth variety represented by workstation clusters. NKS methods fill this niche.

A Newton-Krylov-Schwarz (NKS) method combines a Newton-Krylov (NK) method such as nonlinear GMRES [1], with a Krylov-Schwarz (KS) method, such as additive Schwarz [3]. The key linkage is provided by the Krylov method, in this case the restarted form of GMRES. From a computational point of view, the most important characteristic of a Krylov method for the linear system $Au = f$ is that information about the matrix A needs to be accessed only in the form of matrix-vector products in a small number (relative to the dimension of the matrix) of directions. NK methods are suited for nonlinear problems in which it is unreasonable to compute or store a true Jacobian. However, if the Jacobian A is ill-conditioned, the Krylov method requires an unacceptably large number of iterations and must be preconditioned. It is in the choice of preconditioning where the battle for low computational cost and scalable parallelism is usually won or lost.

In KS methods, the preconditioning is introduced on a subdomain-by-subdomain basis, which provides good data locality for parallel implementations over a range of granularities, and allows significant architectural adaptivity. With an emphasis on operation count complexity and parallel efficiency, Schwarz is usually employed with very modest subdomain overlap and in a two-level form, in which a small global problem is solved together with

* Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 and ICASE, NASA Langley Research Center, Hampton, VA 23681. *keyes@icase.edu.* This work was supported in part by NSF grant ECS-9527169 and by contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering.

the local subdomain problems at each iteration. Mathematically, if $Au = f$ arises as the linearized correction step of a discretized PDE computation, Schwarz operates by:

1. Decomposing the space of the solution u: $\mathcal{U} = \sum_k \mathcal{U}_k$;
2. Finding the restriction of A to each $\mathcal{U}_k$: $A_k = R_k A R_k^T$, for some restriction operators $R_k : \mathcal{U} \rightarrow \mathcal{U}_k$ and extension operators $R_k^T : \mathcal{U}_k \rightarrow \mathcal{U}$;
3. Preconditioning with the A_k^{-1}, where the inverse of A_k is well defined within the k^{th} subspace.

In Schwarz-style domain decomposition, the subspace $\mathcal{U}_k$ corresponding to subdomain k is the span of nodal basis or other expansion functions with support over the subdomain. A practical Schwarz preconditioner is

$$B^{-1} \equiv \sum_k R_k^T (\tilde{A}_k)^{-1} R_k, \tag{1}$$

where $\tilde{A}_k$ is a convenient approximation to $A_k \equiv R_k A R_k^T$. In this abstract, $\tilde{A}_k$ is usually an incomplete LU (ILU) factorization of A_k, with modest fill permitted. For $k = 1, 2, \ldots$, the R_k and R_k^T are simply gather and scatter operators, respectively, one for each subdomain with small overlap between the subdomains. For an optional $k = 0$ term corresponding to the coarse space, R_0 represents a full-weighting restriction operator in the sense of multigrid, and R_0^T is the corresponding prolongation. Neither A nor B^{-1} is assembled globally. Rather, when their action on a vector is needed, a processor governing each subdomain executes local operations, after receiving a thin buffer of data required from its neighbors to complete stencil operations on the boundary of the subdomain. For the assembly and solution of the coarse-grid component of the preconditioner, data exchanges further than nearest neighbor must generally occur.

The NKS technique is compared in this abstract against a defect correction algorithm common to many implicit codes. The objective of either algorithm is to solve the steady-state conservation equations $f(u) = 0$ through the pseudo-transient form $\frac{\partial u}{\partial t} + f(u) = 0$, where the time derivative is approximated by backwards differencing, with a time step that ultimately approaches infinity. A standard defect correction approach employs an accurate right-hand side residual discretization, $f_{high}(u)$, and a convenient left-hand side Jacobian approximation, $J_{low}(u)$, based on a low-accuracy residual $f_{low}(u)$, to compute a sequence of corrections, $\delta u \equiv u^{n+1} - u^n$. Computational short-cuts are employed in the creation of the left-hand side matrix, which may, for instance, be stabilized by a degree of first-order upwinding that would not be acceptable in the discretization of the residual itself.

The so-called "defect" is $f_{high}(u) - f_{low}(u)$, and the nonlinear defect correction scheme to drive $f_{high}(u)$ to zero is to solve approximately for u^{n+1} in $f_{low}(u^{n+1}) = f_{low}(u^n) - f_{high}(u^n)$, which may be linearized as

$$J_{low}(u^n)\delta u = -f_{high}(u^n). \tag{2}$$

In the case of pseudo-transient computations, the approximate Jacobian J_{low} is based on a low-accuracy residual: $J_{low} = \frac{D}{\delta t} + \frac{\partial f_{low}}{\partial u}$, where D is a scaling matrix. It is required either to solve with J_{low}, itself, or with some further algebraic or parallel approximation, $\tilde{J}_{low}$. Inconsistency between the left- and right-hand sides prevents the use of large time steps, δt, and prevents (2) from being a true Newton method.

A Newton-Krylov approach employs a (nearly) consistent left-hand side obtained by directionally differencing the actual residual, f_{high}:

$$J_{high}(u^n)\ \delta u = -f_{high}(u^n), \tag{3}$$

in which the action of J_{high} on a vector is obtained through directional differencing, for instance, $J_{high}(u^n)v \approx \frac{1}{h}[f_{high}(u^n + hv) - f_{high}(u^n)]$, where h is a small parameter. The operators on both sides of (3) are based on consistent high-order discretizations; hence time steps can be advanced to values as large as linear conditioning permits, recovering a true Newton method in the limit.

Preconditioning (3) by $\tilde{J}_{low}$, for instance on the left, as in

$$(\tilde{J}_{low})^{-1} J_{high}(u^n)\ \delta u = -(\tilde{J}_{low})^{-1} f_{high}(u^n), \tag{4}$$

shifts the inconsistency from the nonlinear to the linear aspects of the problem. This should be contrasted with the customary preconditioned form of (2),

$$(\tilde{J}_{low})^{-1} J_{low}(u^n)\ \delta u = -(\tilde{J}_{low})^{-1} f_{high}(u^n). \tag{5}$$

At this level of abstraction, it is not clear which is better — many nonlinear steps with cheap subiterations (5), or a few nonlinear steps with expensive subiterations (4). Execution time comparisons are more practical arbiters than are rates of convergence for the steady-state residual norm, but running times are sensitive to parametric tuning as well as to architectural parameters. We present a comparison of (4) and (5) in Section 3.

A more comprehensive set of comparisons of this type, comparing (4), (5), and

$$(\tilde{J}_{high})^{-1} J_{high}(u^n)\ \delta u = -(\tilde{J}_{high})^{-1} f_{high}(u^n) \tag{6}$$

may be found in [4]. Of course, (6) relies on possessing the full high-order Jacobian, and is not a matrix-free method.

2. Parallel Scalability of Krylov-Schwarz. Practical scalable parallelism is one of the major motivations for research on and implementation of domain decomposition methods in CFD. We loosely call an algorithm/architecture combination "scalable" if its parallel efficiency is constant asymptotically, in any of several coordinated limits of discrete problem size n and parallel granularity p.

For an iterative numerical method, in which the total execution time $T(n,p)$ is the product of an iteration count $I(n,p)$ with an average cost-per-iteration $C(n,p)$, it is useful to separate the parallel efficiency into two factors: numerical efficiency and implementation efficiency. Numerical efficiency η_n measures the degradation of the convergence rate as the problem is scaled, and implementation efficiency η_i measures the degradation in the cost per iteration as the problem is scaled. For instance, we may take $\eta_n \equiv I(n,1)/I(n,p)$ and $\eta_i \equiv C(n,1)/[p \cdot C(n,p)]$. The numerical efficiency is usually very difficult to predict for a nonlinear problem, particularly when refining the grid (increasing n) resolves new physics. However, for certain domain decomposition methods applied to model linear problems with smooth solutions, the relative numerical efficiency $I(n_1,p_1)/I(n_2,p_2)$, with $p_2 > p_1$ and $n_1/p_1 = n_2/p_2$, can be proved to be 100% asymptotically.

The proof relies on the link between the rate of convergence of Krylov methods and the condition number of the (preconditioned) operator $B^{-1}A$, and on the link between the condition number and the extremal eigenvalues in the symmetric case, which can be estimated by Rayleigh quotients. Upper and lower bounds on the condition number of $B^{-1}A$ may be constructed that are independent of the mesh cell diameter h and the subdomain diameter H, or that depend only upon their ratio. In turn, h and H can be inversely related to n and p in simple problems. The theory, which has evolved over a decade to cover nonsmooth, nonsymmetric and indefinite problems, as well as nonnested spaces, is

TABLE 1
Scalability results – Poisson problem

	Fixed Size				Fixed Mem./proc.			
# proc.	# cells	its.	sec.	sec./it.	# cells	its.	sec.	sec./it.
1	262,144	1*	183.5	183.5	16,384	1*	8.8	8.8
4	262,144	8	325.7	40.7	65,536	7	58.4	8.3
16	262,144	12	96.7	8.1	262,144	12	96.7	8.1
64	262,144	12	17.6	1.5	1,048,576	11	91.2	8.3

digested among other places in [2] and [8]. Before presenting parallel CFD results, we illustrate the performance achievable by such methods on contemporary parallel systems.

A message-passing code for the Poisson problem on a unit square was ported to several machines using MPI. With convergence defined as five orders of magnitude reduction in (unpreconditioned) residual, we tested both fixed-size scalability and fixed-memory-per-node scalability on an Intel Paragon with 1, 4, 16, or 64 subdomains, with one subdomain per processor. The results for a fixed-size 512×512 grid are shown in the left side of Table 1. The right side of Table 1 is based on a problem size that grows from 16K to 1M unknowns, with a 128×128 subdomain problem on every processor. The results coincide in the third row.

On each subdomain, a direct FFT-based method is employed, so that only one iteration is required in the uni-processor case. Asymptotically, approximately 12 iterations are required, independent of the granularity. As seen in the column "sec./it." on the right. the implementation efficiency is near perfect in this granularity range. Problems can be solved in constant time as resolution and processing power are increased in proportion. Consulting the fixed size "sec./it." column, we note a super-unitary implementation efficiency – as p increases by a factor of 4, runtime decreases by more than a factor of 4. This is attributable to the cacheing or paging advantages of domain-based array blocking, which are clearly more important than communication effects in this range of n and p.

In general CFD applications, finding a cost-effective coarse-grid operator is not straightforward, and one is often resigned to a Schwarz-preconditioned operator that deteriorates in numerical efficiency as the granularity of the decomposition increases. In such cases, optimizing execution time as a function of granularity is difficult, apart from numerical experimentation.

3. Aerodynamics Application. In this section, we present parallel numerical results for inviscid, subsonic compressible external flow over a two-dimensional multiple-element airfoil using Newton-Krylov-Schwarz modeled by the Euler equations:

$$\nabla \cdot (\rho \mathbf{v}) = 0 \tag{7}$$

$$\nabla \cdot (\rho \mathbf{v}\mathbf{v} + pI) = 0 \tag{8}$$

$$\nabla \cdot ((\rho e + p)\mathbf{v}) = 0 \tag{9}$$

where ρ is the fluid density, $\mathbf{v}$ the velocity, p the pressure, and e the specific total energy, together with the ideal gas law, $p = \rho(\gamma - 1)(e - |\mathbf{v}|^2/2)$, where γ is the ratio of specific heats.

The problem of inviscid incompressible flow around a two-dimensional four-element airfoil in landing configuration was studied in terms of convergence rate and parallel performance in [9], and the same code was converted to NKS form for the present study. The

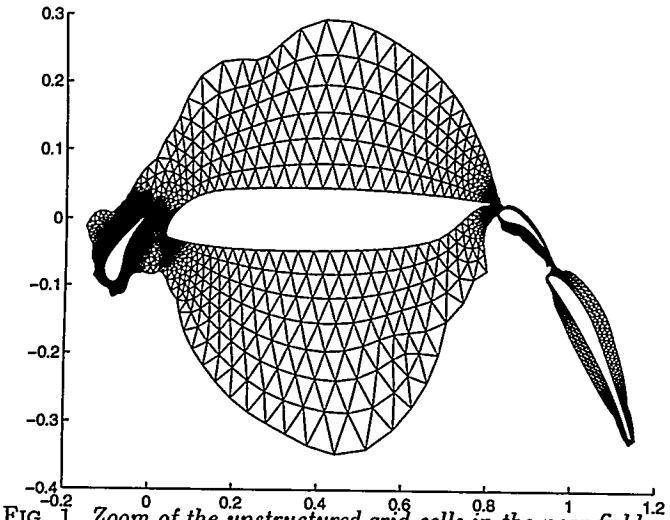


FIG. 1. *Zoom of the unstructured grid cells in the near field.*

details of the discretization are left to the original reference. From [9] we consider the vertex-based discretization with a first-order Roe scheme on the left (out of which we form $\tilde{J}_{low}^{-1}$), and a second-order Roe scheme on the right (which defines f_{high}). The flow is subsonic (Ma = 0.2), with an angle of attack of 5°. Adaptively placed unstructured grids of approximately 6,000 and 16,000 vertices were decomposed into from 1 to 128 load-balanced subdomains, including all power-of-two granularities in between. We report below on the problem of 6,019 vertices, with four degrees of freedom per vertex (giving 24,076 as the algebraic dimension of the discrete problem). This is certainly small by parallel computational standards, though it is probably reasonably adequate in two dimensions from a physical modeling point of view, since the unstructured grid is not restricted to quasi-uniformity, and mesh cells are concentrated into small regions between the airfoils requiring the greatest refinement. The clustering can be seen in Fig. 1, which shows just a near field subset of the grid. (The grid recedes into the far field with smoothly increasing cell sizes. If the entire grid is is scaled to the page size, the flaps are too small to be visible.)

Figure 2 compares the convergence histories of the defect correction and NKS solvers, over a range of time sufficient to that permit the reduction of the residual of the NKS method to drop to within an order of magnitude of ε_{mach}. Both solvers utilize a residual-adaptive setting of the CFL number (related to the size of the time step δt in the pseudo-transient code), known as "switched evolution/relaxation" (SER) [6]. Starting from some small initial CFL number, CFL is adaptively advanced according to:

$$\mathrm{CFL}^{l+1} = \mathrm{CFL}^{l} \cdot \frac{||f(u)^{l-1}||}{||f(u)^{l}||}.$$

As $||f(u^l)|| \to 0$, $\delta t \to \infty$. Since convergence is not generally monotonic in $||f(u)||$, CFL may also adaptively decrease, and it should be ratcheted away from too large a relative decrease, as well.

Both solvers use the same Schwarz preconditioner, namely one-cell overlap and point-block ILU(0) in each subdomain. NKS is clearly superior to defect correction in convergence rate, though the cost per iteration is sufficiently high that defect correction is faster in execution time up to a modest residual reduction. (The cross-over point in the right plot is at about a reduction of 10^4 of the initial residual. A polyalgorithm, initially defect correction then switched to NKS when defect correction prohibits fast growth in CFL, may ultimately

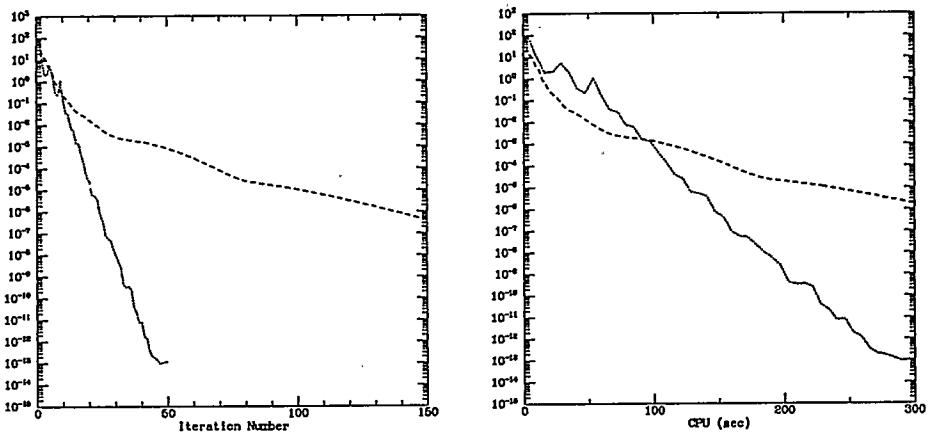


FIG. 2. *Norm of steady-state residual vs. iterations (left) and vs. execution time on 32 nodes of the Intel Paragon (right) for the defect correction scheme (dashed), and the NKS method (dotted).*

be much faster than either pure algorithm exclusively, as demonstrated for a related problem in [7], and as found in preliminary experiments for the present problem.) The asymptotic convergence rate is shown to be linear, since the Newton correction iterations were truncated well above the tolerances necessary to guarantee superlinear or quadratic convergence. Improving the constant in linear convergence is reason enough to use the matrix-free split discretization method (3). Table 2 compares the performance of the NKS version of the solver across three doublings of the processor force of the Intel Paragon for this fixed-size problem. The second-order evaluation of fluxes in $f_{high}(u)$ requires that first conserved variables, and later their fluxes, be communicated across subdomain boundaries each time the routine to evaluate the nonlinear residual is called. This imposes an extra communication burden per iteration on the matrix-free NKS solver, relative to a method that explicitly stores the elements of the Jacobian. Nevertheless, for residual norm reductions of more than a few orders of magnitude, the parallelized NKS solver is faster than the parallelized defect correction solver. The number of subdomains matches the number of processors, so convergence rate of the preconditioned system degrades slowly with increasing granularity, as coupling is lost in the preconditioner. However, the number of Krylov vectors per Newton iteration is bounded (at 2 restart cycles of 25 each), so the data translates directly to parallelization efficiency of the truncated Newton method.

In this example, no coarse grid is used, but [9] compares the defect correction form of the algorithm with and without a coarse grid. The coarse grid appears multiplicatively rather than additively, as in (1). The restriction operator consists of summing subdomain boundary fluxes and the prolongation operator is essentially piecewise constant subdomain extension followed by a boundary relaxation process. On the original platform of the Intel iPSC/2, the convergence rate advantage of the coarse grid is nearly completely cancelled by the sequential bottleneck. The coarse grid aspect of the preconditioner demands further attention.

4. Conclusions and Related Extensions. A variety of CFD applications are (or have inner) nonlinear elliptically-dominated problems amenable to solution by NKS algorithms, which are characterized by low storage requirements (for an implicit method) and locally concentrated data dependencies with small overlaps between the preconditioner blocks. The addition of a global coarse grid in the Schwarz preconditioner is often effective,

TABLE 2
Wall-clock performance and relative parallel efficiency for unstructured Euler code on an Intel Paragon.

# proc.	sec./iter.	rel. eff.
4	36.09	(1.00)
8	19.21	0.94
16	10.65	0.85
32	6.25	0.72

where architecturally convenient. A deterrent to the widespread adoption of NKS algorithms is the large number of parameters that require tuning. Each component (Newton, Krylov, and Schwarz) has its own set of parameters, the most important of which, in our experience, is the convergence criterion for the inner Krylov subiterations. In large-scale, poorly preconditioned problems, including the test problems of this abstract, tunings that guarantee quadratic convergence lead to unacceptable inner iteration counts and/or memory consumption. However, the plethora of parameters can be exploited, in principle, to produce optimal tradeoffs in space and time for a given problem class. Though parametric tuning is important to performance, conservative robust choices are not difficult.

The NK technique has been compared with V-cycle multigrid on Euler and Navier-Stokes problems without parallelizing the preconditioning in [5, 7]. For a subsonic unstructured grid example, NK trails multigrid in execution time by a factor of only about 1.5. This penalty can be accepted when it is realized that the NK method has the advantage of doing all of its computation without generation of a family of coarse unstructured grids (which is difficult for three-dimensional unstructured grids). This work has been extended to three-dimensional problems in [7].

REFERENCES

[1] P. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM Journal of Scientific and Statistical Computing, 11 (1990), pp. 450–481.

[2] T. F. CHAN AND T. MATHEW, *Domain decomposition algorithms*, Acta Numerica, (1994), pp. 61–143.

[3] M. DRYJA AND O. B. WIDLUND, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. Report 339, Courant Institute, NYU, 1987.

[4] H. JIANG AND P. A. FORSYTH, *Robust linear and nonlinear strategies for solution of the transonic Euler equations*, Computers & Fluids, 24 (1995), pp. 753–770.

[5] D. E. KEYES, *Aerodynamic applications of Newton-Krylov-Schwarz solvers*, in Proceedings of the 14th International Conference on Numerical Methods in Fluid Dynamics, R. Narasimha, ed., New York, 1995, Springer Verlag.

[6] W. MULDER AND B. V. LEER, *Experiments with implicit upwind methods for the Euler equations*, Journal of Computational Physics, 59 (1985), pp. 232–246.

[7] E. J. NIELSEN, R. W. WALTERS, W. K. ANDERSON, AND D. E. KEYES, *Application of Newton-Krylov methodology to a three-dimensional unstructured Euler code*, Tech. Report 95-1733, AIAA, 1995.

[8] B. F. SMITH, P. E. BJORSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multilevel Algorithms for Elliptic Partial Differential Equations*, Cambridge Univ. Press, Cambridge, 1996.

[9] V. VENKATAKRISHNAN, *Parallel implicit unstructured grid Euler solvers*, AIAA Journal, 32 (1994), pp. 1985–1991.

Adaptive Parallel Multigrid for Euler and incompressible Navier-Stokes equations

Ulrich Trottenberg

(Joint work with: K. Oosterlee, H. Ritzdorf, A, Schacller, H. Schwichtenberg, B. Steckel, K. Stacben, G. Umlauf, J. Wu)

The combination of

- very efficient solution methods (Multigrid)
- adaptivity

and

- parallelism (distributed memory)

clearly is absolutely necessary for future oriented numerics but still regarded as extremely difficult or even unsolved.

We show that very nice results can be obtained for real life problems. Our approach is straightforward (based on "MLAT"). But, of course, reasonable refinement and load-balancing strategies have to be used. Our examples are 2D, but 3D is on the way.

Domain Decomposition Methods for Mortar Finite Elements

Olof Widlund
widlund@WIDLUND.CS.NYU.EDU

In the last few years, domain decomposition methods, previously developed and tested for standard finite element methods and elliptic problems, have been extended and modified to work for mortar and other nonconforming finite element methods. A survey will be given of work carried out jointly with Yves Achdou, Mario Casarin, Maksymilian Dryja and Yvon Maday. Results on the p- and h-p-version finite elements will also be discussed.

Topic:	*Session Chair:*	*Room B*
Krylov Methods	*Anne Greenbaum*	

8:00 - 8:30	V. Druskin	Extended Krylov Subspaces Approximations of Matrix Functions; Application to Computational Electromagnetics
8:30 - 9:00	D. Sorensen	Krylov Subspace Methods for Computation of Matrix Functions
9:00 - 9:30	T. Tamarchenko	Application of Spectral Lanczos Decomposition Method to Large Scale Problems Arising Geophyics
9:30 - 10:00	A. Greenbaum	Some Uses of the Symmetric Lanczos Algorithm and Why it Works!

Extended Krylov subspaces approximations of matrix functions. Application to computational electromagnetics.

Vladimir Druskin,* Leonid Knizhnerman + and Ping Lee*.

There is now a growing interest in the area of using Krylov subspace approximations to compute the actions of matrix functions. The main application of this approach is the solution of ODE systems, obtained after discretization of partial differential equations by method of lines. In the event that the cost of computing the matrix inverse is relatively inexpensive, it is sometimes attractive to solve the ODE using the extended Krylov subspaces, originated by actions of both positive and negative matrix powers. Examples of such problems can be found frequently in computational electromagnetics.

In this presentation, we introduce an economical Gram-Schmidt orthogonalization on the extended Krylov subspaces of a symmetric matrix. An error bound for a family of problems arising from the elliptic method of lines (i.e. the matrix square root and its stable exponentials) is derived. The bound shows that, for the same approximation quality, the diagonal variant of the extended subspace requires about the square root of the dimension of the standard Krylov subspaces using only positive or negative matrix powers.

Two applications arising from geophysical electromagnetics, one to the solution of a 2.5-D elliptic problem for direct current potential and another for 3-D eddy-current problem in conductive media attest to a computational efficiency of the method.

* Schlumberger-Doll Research, Old Quarry Road,
Ridgefield, CT 06877-4108,
+ Central Geophysical Expedition, Narodnogo Opolcheniya St.,
House 40, Bldg. 3, Moscow 123298, Russia

Applications of Implicit Restarting in Optimization and Control Dan Sorensen

Dan Sorensen
Dept. Computational and Applied Mathematics
Rice University
Houston, TX 77251-1892

Implicit restarting is a technique for combining the implicitly shifted QR mechanism with a k-step Arnoldi or Lanczos factorization to obtain a truncated form of the implicitly shifted QR-iteration suitable for large scale eigenvalue problems. The software package ARPACK based upon this technique has been successfully used to solve large scale symmetric and nonsymmetric (generalized) eigenvalue problems arising from a variety of applications.

Recently, the implicit restarting technique has been applied to problems in control and optimization. The technique has been generalized to provide an implicit restarting technique for the nonsymmetric two sided Lanczos process. This mechanism is used to obtain stable reduced models for state space control systems. Implicit restarting has also found application in the numerical solution of large scale trust region subproblem: Minimize a quadratic function subject to an ellipsoidal constraint.

This talk will survey the applications in control and optimization.

APPLICATION OF SPECTRAL LANCZOS DECOMPOSITION METHOD TO LARGE SCALE PROBLEMS ARISING GEOPHYSICS

T. Tamarchenko
Western Atlas Logging Services
10201 Westheimer
Houston, TX 77042, USA
E-mail: tanya.tamarchenko@waii.com

This paper presents an application of Spectral Lanczos Decomposition Method (SLDM) to numerical modeling of electromagnetic diffusion and elastic waves propagation in inhomogeneous media. SLDM approximates an action of a matrix function as a linear combination of basis vectors in Krylov subspace.

I applied the method to model electromagnetic fields in three-dimensions and elastic waves in two dimensions. The finite-difference approximation of the spatial part of differential operator reduces the initial boundary-value problem to a system of ordinary differential equations with respect to time. The solution to this system requires calculating exponential and sine/cosine functions of the stiffness matrices.

Large scale numerical examples are in a good agreement with the theoretical error bounds and stability estimates given by Druskin, Knizhnerman, 1987.

Some Uses of the Symmetric Lanczos Algorithm — and Why it Works!

V.L. Druskin* A. Greenbaum† L.A. Knizhnerman‡

January 1, 1996

Abstract

The Lanczos algorithm uses a three-term recurrence to construct an orthonormal basis for the Krylov space corresponding to a symmetric matrix A and a starting vector q_1. The vectors and recurrence coefficients produced by this algorithm can be used for a number of purposes, including solving linear systems $Au = \varphi$ and computing the matrix exponential $e^{-tA}\varphi$. Although the vectors produced in finite precision arithmetic are not orthogonal, we show why they can still be used effectively for these purposes.

The reason is that the 2-norm of the residual is essentially determined by the *tridiagonal matrix* and the next *recurrence coefficient* produced by the finite precision Lanczos computation. It follows that if the same tridiagonal matrix and recurrence coefficient are produced by the exact Lanczos algorithm applied to some other problem, then exact arithmetic bounds on the residual for that problem will hold for the finite precision computation. In order to establish exact arithmetic bounds for the different problem, it is necessary to have some information about the eigenvalues of the new coefficient matrix. Here we make use of information already established in the literature, and we also prove a new result for indefinite matrices.

*Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877-4108

†Courant Institute of Mathematical Sciences, 251 Mercer St., New York, NY 10012.
This work was supported by NSF grant 25968-5375.
‡Central Geophysical Expedition, Narodnogo Opolcheniya St., House 40, Bldg. 3, Moscow 123298, Russia

Topic: *CFD*	*Session Chair:* *TBA*	*Room C*
8:00 - 8:30	X. Zheng	Multigrid Solution of Incompressible Turbulent Flows by Using Two-Equation Turbulence Models
8:30 - 9:00	S. Kumar	Nonlinear Krylov Acceleration of Reacting Flow Codes
9:00 - 9:30	M.D. Tidriri	Schwarz-based Algorithms for Compressible Flows
9:30 - 10:00	C. Liao	Multilevel Local Refinement and Multigrid Methods for 3-D Turbulent Flow

Multigrid Solution of Incompressible Turbulent Flows By using Two-Equation Turbulence Models

X. Zheng, C. Liu
Front Range Scientific Computations, INC., Denver, Colorado
C. H. Sung
David Taylor Model Basin, Bethesda, Maryland

Most of practical flows are turbulent. From the interest of engineering applications, simulation of realistic flows is usually done through solution of Reynolds-averaged Navier-Stokes equations and turbulence model equations. It has been widely accepted that turbulence modeling plays a very important role in numerical simulation of practical flow problem, particularly when the accuracy is of great concern. Among the most used turbulence models today, two-equation models appear to be favored for the reason that they are more general than algebraic models and affordable with current available computer resources. However, investigators using two-equation models seem to have been more concerned with the solution of N-S equations. Less attention is paid to the solution method for the turbulence model equations. In most cases, the turbulence model equations are loosely coupled with N-S equations, multigrid acceleration is only applied to the solution of N-S equations due to perhaps the fact the turbulence model equations are source-term dominant and very stiff in sublayer region.

In this paper, a multigrid method is developed to solve the two-equation turbulence models as well as the N-S equations. These two sets of equations are solved by using a strongly-coupled time marching method. Two popular two-equation models, k-ω and k-ϵ, are discussed in this work. A point-implicit technique is developed to improve the efficiency of the solution, and more importantly to alleviate the stiffness of the governing equations exhibited in near wall region. Treatments of important source terms of turbulence model through the multigrid process is tested and analyzed.

The method is first tested by applying to the classic flat plate boundary layer flow. The efficiency, robustness and accuracy of the method are demonstrated by the calculation of incompressible flow around a underwater vehicle model at a Renolds number more that ten million.

Nonlinear Krylov Acceleration of Reacting Flow Codes

S. Kumar, R. Rawat, P. Smith
Department of Chemical and Fuels Engineering
University of Utah
Salt Lake City, Utah 84112

M. Pernice
Utah Supercomputing Institute
University of Utah
Salt Lake City, Utah 84112

We are working on computational simulations of three-dimensional reactive flows in applications encompassing a broad range of chemical engineering problems. Examples of such processes are coal (pulverized and fluidized bed) and gas combustion, petroleum processing (cracking), and metallurgical operations such as smelting. These simulations involve an interplay of various physical and chemical factors such as fluid dynamics with turbulence, convective and radiative heat transfer, multiphase effects such as fluid-particle and particle-particle interactions, and chemical reaction.

The governing equations resulting from modeling these processes are highly nonlinear and strongly coupled, thereby rendering their solution by traditional iterative methods (such as nonlinear line Gauss-Seidel methods) very difficult and sometimes impossible. Hence we are exploring the use of nonlinear Krylov techniques (such as GMRES and Bi-CGSTAB) to accelerate and stabilize the existing solver. This strategy allows us to take advantage of the problem-definition capabilities of the existing solver. The overall approach amounts to using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) method and its variants as nonlinear preconditioners for the nonlinear Krylov method. We have also adapted a backtracking approach for inexact Newton methods to damp the Newton step in the nonlinear Krylov method.

This will be a report on work in progress. Preliminary results with nonlinear GMRES have been very encouraging: in many cases the number of line Gauss-Seidel sweeps has been reduced by about a factor of 5, and increased robustness of the underlying solver has also been observed.

Schwarz-based algorithms for compressible flows

M. D. Tidriri *

ICASE
MS 132C, NASA-LaRC, Hampton, VA 23681-0001

1 Methodology

To compute steady compressible flows one often uses an implicit discretization approach which leads to a large sparse linear system that must be solved at each time step. In the derivation of this system one often uses a defect-correction procedure, in which the left-hand side of the system is discretized with a lower order approximation than that used for the right-hand side. This is due to storage considerations and computational complexity, and also to the fact that the resulting lower order matrix is better conditioned than the higher order matrix. The resulting schemes are only moderately implicit. In the case of structured, body-fitted grids, the linear system can easily be solved using approximate factorization (AF), which is among the most widely used methods for such grids. However, for unstructured grids, such techniques are no longer valid, and the system is solved using direct or iterative techniques. Because of the prohibitive computational costs and large memory requirements for the solution of compressible flows, iterative methods are preferred. In these defect-correction methods, which are implemented in most CFD computer codes, the mismatch in the right and left hand side operators, together with explicit treatment of the boundary conditions, lead to a severely limited CFL number, which results in a slow convergence to steady state aerodynamic solutions. Many authors have tried to replace explicit boundary conditions with implicit ones (see for instance [11], [7], and [6]). Although they clearly demonstrate that high CFL numbers are possible, the reduction in CPU time is not clear cut.

The investigation of defect-correction procedures based on Krylov methods, together with implicit treatment of the boundary conditions has been done by the author in [10]. In [10] the author has also studied Newton-Krylov matrix-free (see also [1], [8], [9], [2], and [3]) methods combined with mixed discretization in the implicitly defined Jacobian-Preconditioner. The preconditioner based on incomplete factorizations studied in [10], is difficult to parallelize efficiently. The focus in this work is on

*email:tidriri@icase.edu. This work was supported by the National Aeronautics and Space Administration under NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering.

the developement of algorithms that are suitable for parallel computing environment. In this case, domain decomposition methods that allow the reduction of the global solution of a given problem to the solutions of local subproblems are preferred. We propose, therefore, to combine these methods with the preconditioned Newton-Krylov matrix-free methods developed in [10].

One of the domain decomposition algorithms that has potential applications on parallel computers is the additive Schwarz algorithm [4]. The other Schwarz-based method; the multiplicative Schwarz method [4] can also be used in parallel environment computing by using a coloring process. The proposed algorithm is, therefore, to combine the Newton-Krylov matrix-free methods with the Schwarz-based methods. The combination of Newton-Krylov matrix-free with domain decomposition methods was first introduced by the author in [8] and [9]. More precisely the author has combined Newton-Krylov matrix-free method with the Domain Decomposition Time Marching Algorithm that was introduced by Le Tallec and Tidriri in [5] (see also [8] and [9]).

2 Results

To test the proposed algorithm we consider a NACA0012 steady transonic airfoil at an angle of attack of 1.25 degrees and a freestream Mach number of 0.8. The mesh we use is composed of 4096 cells. To illustrate the overall benefit of the combination of the Schwarz-based algorithms with the Newton-Krylov matrix-free methods as compared to their combination with the more standard defect-correction procedures, using implicit boundary conditions, we present in figure 1 the curv presenting the logarithm of the nonlinear steady-state residual versus the CPU time. This curv corresponds to the particular Schwarz-based method: the additive Schwarz algorithm using the mesh described above. This corresponds also, to the particular 8×8 subdomain decomposition. In all computations performed herein the solution obtained agrees with the standard one. All Those calculations are performed on the same Sparc10 machine. The relative tolerance in the solution of the linear system is 10^{-3} for the preconditioned Krylov methods (ILU/GMRES). The steady state regime is declared when the nonlinear residual norm reaches a value of (or less than) 10^{-5}.

References

[1] P. N. Brown and Y. Saad, *Hybrid Krylov Methods for Nonlinear Systems of Equations*, SIAM J. Sci. Stat. Comp. **11**(1990), 450–481.

[2] X.-C. Cai, W. D. Gropp, D. E. Keyes and M. D. Tidriri, "Parallel implicit methods for aerodynamics," Seventh International Conference on Domain Decomposition Methods for Partial Differential Equations, D. Keyes, J. Xu, eds., AMS, 1994.

[3] X.-C. Cai, W. D. Gropp, D. E. Keyes and M. D. Tidriri, "Newton-Krylov-Schwarz Methods in CFD," Proceedings of the International Workshop on the Navier-Stokes Equations, Notes in Numerical Fluid Mechanics, R. Rannacher, eds. Vieweg Verlag, Braunschweig, 1994.

[4] M. Dryja and O. Widlund, *Towards a Unified Theory of Domain Decomposition Algorithms for Elliptic Problems,* in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, 1989.

[5] P. Le Tallec and M. D. Tidriri, *Convergence of domain decomposition algorithms with full overlapping for the advection-diffusion problems,* INRIA Research Report RR-2435, Oct. 1994, also submitted to Math. Comp.

[6] M.-S. Liou, and B. Van Leer, *Choice of Implicit and Explicit Operators for the Upwind Differencing Method,* AIAA Paper, AIAA-88-0624 (1988).

[7] W. T. Thomkins, Jr. and R. H. Bush, *Boundary Treatments for Implicit Solutions to Euler and Navier-Stokes Equations,* J. Comp. Phys. **48.** (1982), 302–311.

[8] M. D. Tidriri, *Coupling of different models and different approximations in the computation of external flows,* PhD thesis, Univ. of Paris XI, 1992.

[9] M. D. Tidriri, *Domain Decompositions for Compressible Navier-Stokes Equations with different discretizations and formulations,* J. Comp. Phys. July, 1995.

[10] M. D. Tidriri, *Krylov Methods for Compressible Flows,* ICASE-TR June, 1995.

[11] H. C. Yee, R. M. Beam, and R. F. Warming *Boundary Approximations for Implicit Schemes for One-Dimensional Inviscid Equations of Gasdynamics,* AIAA Journal, Vol.20, NO.9, September 1982.

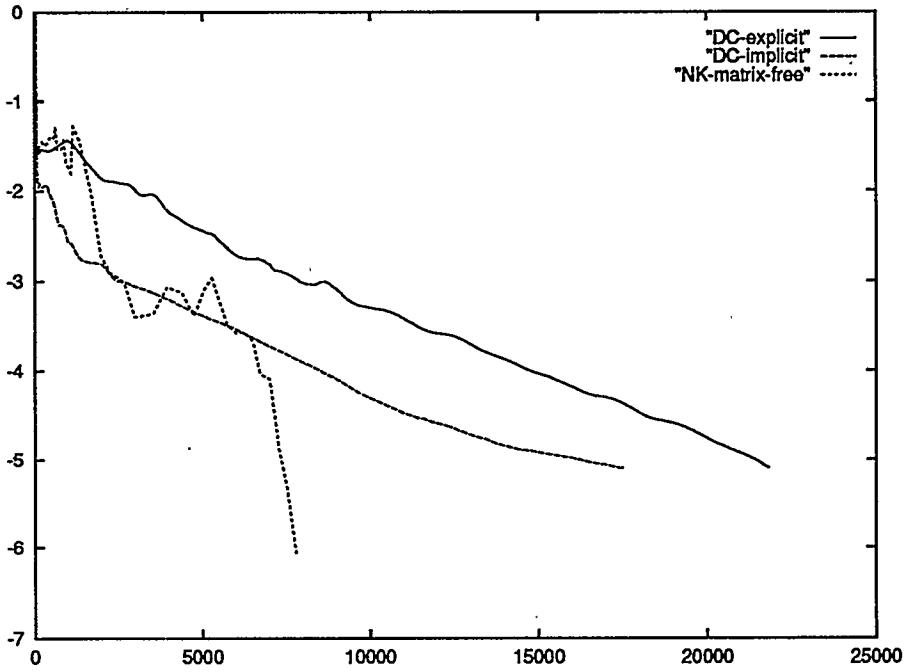


Figure 1: Steady-state residual versus CPU time (in seconds) for steady transonic flow at convergence for the 8×8 decompositions, employing the additive Schwarz algorithm combined with defect-correction procedures with explicit (DC-explicit) and implicit (DC-implicit) boundary conditions, and with Newton-Krylov matrix-free (NK-matrix-free) methods.

Multilevel Local Refinement and Multigrid Methods for 3-D Turbulent Flow

C. Liao and C. Liu
Center for Computational Mathematics, UCD
Campus Box 170, P.O. Box 173364
Denver, CO 80217-3364

C.H. Sung and T. T. Huang
David Taylor Model Basin
Carderock Division, NSWC
Bethesda, Maryland

Abstract

A numerical approach based on multigrid, multilevel local refinement, and preconditioning methods for solving incompressible Reynolds-averaged Navier-Stokes equations is presented. 3-D turbulent flow around an underwater vehicle is computed. 3 multigrid levels and 2 local refinement grid levels are used. The global grid is $24 \times 8 \times 12$. The first patch is $40 \times 16 \times 20$ and the second patch is $72 \times 32 \times 36$. 4th order artificial dissipation are used for numerical stability. The conservative artificial compressibility method are used for further improvement of convergence. To improve the accuracy of coarse/fine grid interface of local refinement, flux interpolation method for refined grid boundary is used. The numerical results are in good agreement with experimental data. The local refinement can improve the prediction accuracy significantly. The flux interpolation method for local refinement can keep conservation for a composite grid, therefore further modify the prediction accuracy.

Topic: *Doman Decomp.*	*Session Chair:* *Olof Widlund*	*Room A*

10:30 - 11:00	X. Feng	A Mixed Finite Element Domain Decomposition Method for Solving Nearly Elastic Wave Equations in the Frequency Domain
11:00 - 11:30	A. Jemcov	Representation of Discrete Steklov-Poincare Operator Arising in Domain Decomposition Methods in Wavelet Basis
11:30 - 12:00	J. Xu	Simplified Approach to Some Nonoverlapping Domain Dcomp. Methods

A MIXED FINITE ELEMENT DOMAIN DECOMPOSITION METHOD FOR NEARLY ELASTIC WAVE EQUATIONS IN THE FREQUENCY DOMAIN

XIAOBING FENG†

ABSTRACT. A non–overlapping domain decomposition iterative method is proposed and analyzed for mixed finite element methods for a sequence of noncoercive elliptic systems with radiation boundary conditions. These differential systems describe the motion of a nearly elastic solid in the frequency domain. The convergence of the iterative procedure is demonstrated and the rate of convergence is derived for the case when the domain is decomposed into subdomains in which each subdomain consists of an individual element associated with the mixed finite elements. The hybridization of mixed finite element methods plays a important role in the construction of the discrete procedure.

§1. Introduction. Domain decomposition (DD) methods have been studied extensively and become very attractive for their parallelism and flexibility (cf. [3], [5], [14], [16], [17] and the references therein). In domain decomposition methods, a domain over which the problem is defined is partioned into subdomains, then the original problem is decomposed into a number of subdomain problems which could be solve in parallel. Another advantage of dividing the whole domain problem into subdomain problems is that even on sequential computers one can approximate the parts of the solution with greater independence. The major difficulties with such domain decomposition procedures are to transfer information between subdomains and to piece the subdomain solutions together into a reasonable approximation of the true solution to the given problem.

The objective of this paper is to develop a non–overlapping domain decomposition method based on mixed finite element methods for the nearly elastic wave equations in the frequency domain. The systems considered are elliptic but noncoercive, which have the similar characteristics to the Helmhotz equation. One motivation for developing domain decomposition iterative method for the problem is that the classical relaxation methods such as Jocobi and SOR methods are not convergent for the problem, the other motivation is that the method can be very naturally implemented on a parallel computer by assigning each subdomain to its own processor. The main point is to use Robin type boundary conditions to pass information between subdomains, on the other hand, to realize this idea to the mixed finite element equations poses the main difficulty for the analysis and implementation.

The iterative procedures given in this paper is closely related to one developed by Després [7] for the Helmhotz equation, and to the procedure developed by Douglas, Paes Leme, Roberts and Wang [8]. The hybridization of mixed finite element methods is strongly used in our iterative procedure. A related procedure for nearly elastic wave equations based on the nonconforming Wilson finite element was developed in [2].

We remark that this paper is a much condensed version of [11], which may be regarded as a long abstract of [11]. The domain decomposition method for the continuous differential problem

1991 *Mathematics Subject Classification.* 65N30, 65N55.

Key words and phrases. mixed finite element methods, domain decomposition, nearly elastic wave equations.

†Department of Mathematics, The University of Tennessee, Knoxville, TN 37996.

is presented in Section 2, its application for solving the mixed finite element approximations to the systems using the the Johnson–Mercier element [13] and Arnold–Douglas–Gupta elements [1] are illustrated briefly in Section 3. Almost all the proofs are omitted, interested readers are referred to [11] for the details.

§2. The DD method for the differential problem. In this paper we consider the following sequence of elliptic systems:

$$-\omega^2 \underset{\sim}{u} - \operatorname{div} \underset{\approx}{\sigma}(\underset{\sim}{u}) = \underset{\sim}{f}, \qquad \text{in } \Omega, \tag{2.1.i}$$

$$\underset{\approx}{\sigma}(\underset{\sim}{u})\underset{\sim}{\nu} + i\omega \underset{\approx}{A}\underset{\sim}{u} = \underset{\sim}{g}, \qquad \text{on } \Gamma = \partial\Omega, \tag{2.1.ii}$$

for each $\omega > 0$. Where Ω is a convex polygonal domain in $\mathbb{R}^N$ for $N = 2, 3$, in particular, we are interested in the case that $\Omega = (0,1)^N$. $\underset{\sim}{\nu}$ denotes the outward normal vector on Γ, $\underset{\sim}{u}$ is the displacement vector in the frequency domain. The stress–strain relation in the frequency domain is described as follows:

$$\underset{\approx}{\sigma} = \lambda \operatorname{tr}(\underset{\approx}{\varepsilon}(\underset{\sim}{u}))\underset{\approx}{I} + 2\mu \underset{\approx}{\varepsilon}(\underset{\sim}{u}), \quad \text{in } \Omega, \tag{2.2.i}$$

$$\underset{\approx}{\varepsilon}(\underset{\sim}{u}) = \frac{1}{2}(\nabla \underset{\sim}{u} + \nabla \underset{\sim}{u}^{\perp}), \qquad \text{in } \Omega, \tag{2.2.ii}$$

$$\lambda = \lambda_r + i\lambda_i, \quad \mu = \mu_r + i\mu_i, \tag{2.2.iii}$$

where $\underset{\approx}{I}$ denotes the $N \times N$ identity matrix. The coefficients λ_r and μ_r are known as the Lamé constants for the material. Also, it is assumed that λ_i and μ_i are strictly positive and that $\lambda_i << \lambda_r$ and $\mu_i << \mu_r$. The coefficients λ_i and μ_i are not measurable directly but are related to other parameters measuring attenuation (cf. [15]). Finally, $\underset{\sim}{f}$ denotes the source vector and $\underset{\approx}{A}$ is a given $N \times N$ positive definite constant matrix. The boundary condition (2.1.ii) with $\underset{\sim}{g} = 0$ is a standard first order absorbing boundary condition which allows waves striking normally to the boundary Γ to be completely annihilated ([11], [15]).

We remark that when $\mu_i = \lambda_i = 0$, the material becomes an elastic material and (2.1) is nothing but the Fourier–transformed (in time) equations of the classical elastic wave equations. So the frequency domain formulations for elastic waves are also included in (2.1) and they can be regarded as limiting forms of nearly elastic waves as λ_i and μ_i go to zero.

For any space X, let $\underset{\sim}{X}$ [respectively, $\underset{\approx}{X}$] denote the space of N–vectors [$N \times N$–tensors] with components in X. If X is normed, associated norms are defined by

$$\|\underset{\sim}{v}\|_{\underset{\sim}{X}} = \left(\sum_{j=1}^{N} \|v_j\|_X^2\right)^{\frac{1}{2}}, \qquad \|\underset{\approx}{\tau}\|_{\underset{\approx}{X}} = \left(\sum_{j,k=1}^{N} \|\tau_{jk}\|_X^2\right)^{\frac{1}{2}},$$

and the subspace $\underset{\approx}{X}_s$ of $\underset{\approx}{X}$ consisting of $N \times N$–tensors which are symmetric. We also introduce the following special notations:

$$V = L^2(\Omega), \qquad \underset{\approx}{H} = \underset{\approx}{H}(\operatorname{div}; \Omega)_s = \{\underset{\approx}{\tau} \in \underset{\approx}{L}^2(\Omega)_s : \operatorname{div} \underset{\approx}{\tau} \in \underset{\sim}{V}\}.$$

Notice that in ths paper all functions are complex–valued.

Theorem 2.1 (cf. [2]). *For any given* $\underset{\sim}{f} \in \underset{\sim}{H}^{-1}(\Omega)$ *and* $\underset{\sim}{g} \in \underset{\sim}{H}^{-\frac{1}{2}}(\Gamma)$, *problem (2.1) admits a unique solution* $\underset{\sim}{u} \in \underset{\sim}{H}^{-1}(\Omega)$. *Moreover, if* $\underset{\sim}{f} \in \underset{\sim}{L}^2(\Omega)$ *and* $\underset{\sim}{g} \in \underset{\sim}{L}^2(\Gamma)$, *then* $\underset{\sim}{u} \in \underset{\sim}{H}^{\frac{3}{2}}(\Omega)$.

Applying the trace operator tr to both sides of (2.1.i) and solving for the $\mathrm{tr}(\underset{\approx}{\varepsilon}(\underset{\sim}{u}))$, we see that

$$\mathrm{tr}(\underset{\approx}{\varepsilon}(\underset{\sim}{u})) = \gamma\,\mathrm{tr}(\underset{\sim}{u}), \quad \text{where } \gamma = \begin{cases} \frac{\lambda}{4\mu(\lambda+\mu)}, & \text{for } N=2, \\ \frac{\lambda}{2\mu(3\lambda+2\mu)}, & \text{for } N=3. \end{cases} \tag{2.3}$$

Substituting (2.3) into (2.2) we get

$$\frac{1}{2\mu}\underset{\approx}{\sigma} - \gamma\,\mathrm{tr}(\underset{\approx}{\sigma})\underset{\approx}{I} - \underset{\approx}{\varepsilon}(\underset{\sim}{u}) = 0.$$

So the mixed formulation of (2.1) is defined by seeking $(\underset{\approx}{\sigma}, \underset{\sim}{u}) \in \underset{\approx}{H} \times \underset{\sim}{V}$ such that

$$a(\underset{\approx}{\sigma}, \underset{\approx}{\tau})_\Omega + (\underset{\sim}{u}, \mathrm{div}\,\underset{\approx}{\tau})_\Omega - i\omega^{-1}\langle A^{-1}\underset{\approx}{\sigma}\underset{\sim}{\nu}, \underset{\approx}{\tau}\underset{\sim}{\nu}\rangle_\Gamma = -i\omega^{-1}\langle A^{-1}\underset{\sim}{g}, \underset{\approx}{\tau}\underset{\sim}{\nu}\rangle_\Gamma, \quad \underset{\approx}{\tau} \in \underset{\approx}{H}, \tag{2.4.i}$$

$$(\mathrm{div}\,\underset{\approx}{\sigma}, \underset{\sim}{v})_\Omega + \omega^2(\underset{\sim}{u}, \underset{\sim}{v})_\Omega = -(\underset{\sim}{f}, \underset{\sim}{v})_\Omega, \quad \underset{\sim}{v} \in \underset{\sim}{V}, \tag{2.4.ii}$$

where

$$a(\underset{\approx}{\sigma}, \underset{\approx}{\tau})_\Omega = \int_\Omega [\frac{1}{2\mu}\underset{\approx}{\sigma}\overline{\underset{\approx}{\tau}} - \gamma\,\mathrm{tr}(\underset{\approx}{\sigma})\,\mathrm{tr}(\overline{\underset{\approx}{\tau}})]\,dx.$$

The unique solvability of (2.4) is ensured by the following theorem (cf. [11]).

Theorem 2.2. *Let* $\underset{\sim}{u} \in \underset{\sim}{H}^1(\Omega)$ *be the solution of (2.1), then* $(\underset{\approx}{\sigma}(\underset{\sim}{u}), \underset{\sim}{u}) \in \underset{\approx}{H} \times \underset{\sim}{V}$ *is the unique solution of (2.4). Conversely, if* $(\underset{\approx}{\sigma}, \underset{\sim}{u}) \in \underset{\approx}{H} \times \underset{\sim}{V}$ *is a solution of (2.4), then* $\underset{\sim}{u} \in \underset{\sim}{H}^1(\Omega)$ *and it is the unique weak solution of (2.1).*

Let $\{\Omega_j\}_{j=1}^J$ be a non–overlapping partition of Ω with Lipschitz boundaries $\{\partial\Omega_j\}$. Introduce the notations

$$\Gamma_k = \Gamma_{k0} = \Gamma \cap \partial\Omega_k, \quad \text{and} \quad \Gamma_{kj} = \partial\Omega_k \cap \partial\Omega_j.$$

Let $\underset{\sim}{u}_j$ denote the restriction of the solution $\underset{\sim}{u}$ on Ω_j. It is well–known that $\underset{\sim}{u}_j$ must satisfy the consistency conditions

$$\underset{\sim}{u}_k = \underset{\sim}{u}_j, \qquad \underset{\approx}{\sigma}_k\underset{\sim}{\nu}_k = -\underset{\approx}{\sigma}_j\underset{\sim}{\nu}_j, \quad \text{on } \Gamma_{kj}. \tag{2.5}$$

It is more convenient ([14], [7] and [8]) to replace (2.5) by the following Robin type boundary condition on the interface Γ_{kj}:

$$\underset{\approx}{\sigma}_k\underset{\sim}{\nu}_k + \alpha\underset{\sim}{u}_k = -\underset{\approx}{\sigma}_j\underset{\sim}{\nu}_j + \alpha\underset{\sim}{u}_j, \qquad \text{on } \Gamma_{kj}, \tag{2.6.i}$$

$$\underset{\approx}{\sigma}_j\underset{\sim}{\nu}_j + \alpha\underset{\sim}{u}_j = -\underset{\approx}{\sigma}_k\underset{\sim}{\nu}_k + \alpha\underset{\sim}{u}_k, \qquad \text{on } \Gamma_{kj}, \tag{2.6.ii}$$

where α is a nonzero complex number. In this paper we choose $\alpha = -\alpha_r + i\alpha_i$ with $\alpha_r \geq 0$ and $\alpha_i > 0$. The reason for the restriction will be clear later in Theorem 2.3.

Let $\underset{\approx}{H}_k = \underset{\approx}{H}\,|_{\Omega_k}$ and $\underset{\sim}{V}_k = \underset{\sim}{V}\,|_{\Omega_k}$. Based on the consistency condition (2.6), we define the following iterative algorithm for (2.4) with $g = 0$:

Choose $(\underset{\approx}{\sigma}_k^0, \underset{\sim}{u}_k^0) \in \underset{\approx}{H}_k \times \underset{\sim}{V}_k$ with $\underset{\approx}{\sigma}_k^0 \underset{\sim}{\nu}_k \in \underset{\sim}{L}^2(\Gamma_{kj})$, $k, j = 1, \cdots J$, arbitrarily, then compute $(\underset{\approx}{\sigma}_k^n, \underset{\sim}{u}_k^n) \in \underset{\approx}{H}_k \times \underset{\sim}{V}_k$ for $n \geq 1$ recursively by solving

$$\text{(2.7.i)} \quad a(\underset{\approx}{\sigma}_k^n, \underset{\approx}{\tau})_{\Omega_k} + (\underset{\sim}{u}_k^n, \operatorname{div} \underset{\approx}{\tau})_{\Omega_k} - i\omega^{-1}\langle A^{-1}\underset{\approx}{\sigma}_k^n \underset{\sim}{\nu}_k, \underset{\approx}{\tau}\underset{\sim}{\nu}_k\rangle_{\Gamma_k} - \sum_{j=1}^{J}\langle \underset{\sim}{u}_k^n, \underset{\approx}{\tau}\underset{\sim}{\nu}_k\rangle_{\Gamma_{kj}} = 0, \forall \underset{\approx}{\tau} \in \underset{\approx}{H}_k,$$

$$\text{(2.7.ii)} \quad (\operatorname{div} \underset{\approx}{\sigma}_k^n, \underset{\sim}{v})_{\Omega_k} + \omega^2(\underset{\sim}{u}_k^n, \underset{\sim}{v})_{\Omega_k} = -(\underset{\sim}{f}, \underset{\sim}{v})_{\Omega_k}, \qquad \forall \underset{\sim}{v} \in \underset{\sim}{V}_k,$$

$$\text{(2.7.iii)} \quad \underset{\approx}{\sigma}_k^n \underset{\sim}{\nu}_k + \alpha \underset{\sim}{u}_k^n = -\underset{\approx}{\sigma}_j^{n-1}\underset{\sim}{\nu}_j + \alpha \underset{\sim}{u}_j^{n-1}, \qquad \text{on } \Gamma_{kj}.$$

Remark. Clearly, from Theorems 2.1 and 2.2 we know that the iterate sequence $\{(\underset{\approx}{\sigma}_k^n, \underset{\sim}{u}_k^n)\}$ is well–defined.

The usefulness of this domain decomposition iterative algorithm is demonstrated by the following convergence theorem.

Theorem 2.3. *The solution $(\underset{\approx}{\sigma}_k^n, \underset{\sim}{u}_k^n)$ of (2.7) converges to the solution $(\underset{\approx}{\sigma}, \underset{\sim}{u})$ of (2.4) strongly in $\underset{\approx}{H}_k \times \underset{\sim}{V}_k$ provided that $\alpha_r \geq 0$ and $\alpha_i > 0$ in the parameter $\alpha = -\alpha_r + i\alpha_i$.*

To sketch the idea of the proof, we need to introduce the "pseudo energy"

$$E(\{\underset{\approx}{\pi}, \underset{\sim}{e}\}) = \sum_{k=1}^{J}\sum_{j=1}^{J}\int_{\Gamma_{kj}} |\underset{\approx}{\pi}_k \underset{\sim}{\nu}_k + \alpha \underset{\sim}{e}_k|^2 ds. \tag{2.8}$$

Let $\underset{\sim}{e}_k^n = \underset{\sim}{u}_k^n - \underset{\sim}{u}|_{\Omega_k}$, $\underset{\approx}{\pi}_k^n = \underset{\approx}{\sigma}_k^n - \underset{\approx}{\sigma}|_{\Omega_k}$, and $E^n = E(\{\underset{\approx}{\pi}_j^n, \underset{\sim}{e}_j^n\})$. Then we have

Lemma 2.1 (cf. [11]). *There holds the following identity:*

$$E^{n+1} = E^n - \sum_{k=1}^{J}\Big\{4\alpha_i[\operatorname{Im} a(\underset{\approx}{\pi}_k^n, \underset{\approx}{\pi}_k^n)_{\Omega_k} + \omega^{-1}\langle A^{-1}\underset{\approx}{\pi}_k^n \underset{\sim}{\nu}_k, \underset{\approx}{\pi}_k^n \underset{\sim}{\nu}_k\rangle_{\Gamma_k}] + 4\alpha_r[\omega^2 \|\underset{\sim}{e}_k^n\|_{0,\Omega_k}^2 - \operatorname{Re} a(\underset{\approx}{\pi}_k^n, \underset{\approx}{\pi}_k^n)_{\Omega_k}]\Big\}. \tag{2.9}$$

Lemma 2.1 says that the "pseudo energy" of the error function sequence $\{(\underset{\approx}{\pi}_j^n, \underset{\sim}{e}_j^n)\}$ is strictly decreasing as the iteration number n increases. Using this fact we can show that $(\underset{\approx}{\pi}_j^n, \underset{\sim}{e}_j^n)$ converges to zero in $\underset{\approx}{H}_k \times \underset{\sim}{V}_k$.

§3. The DD method for mixed finite element approximations. To discretize the algorithm (2.7), we are interested in treating the case in which one subdomain equals one finite element of small diameter though larger subdomains are permissible, that is, $\{\Omega_j\}$ is a partition of Ω into individual elements (simplices, rectangles, prisms, tetrahedrons).

Due to the difficulty in constructing the finite element space for the symmetric stress tensor space $\underset{\approx}{H}$, the construction of effective and stable mixed finite element spaces for elasticity problems has proven to be very difficult and has not yet been accomplished in a completely satisfactory manner for plane, especially for three–dimensional elasticity problems (see [1] and [4]

for a discussion on this point). In this section we will focus on the presentation of the application of the domain decomposition algorithm by confining us only to consider the problem in the two spatial dimension.

Let $\underset{\approx}{H}^h \times \underset{\sim}{V}^h$ denote a mixed finite element subspace of $\underset{\approx}{H} \times \underset{\sim}{V}$. Several choices of $\underset{\approx}{H}^h \times \underset{\sim}{V}^h$ are acceptable (cf. [4]). Here we only consider the subspace of Johnson–Mercier [13] and the family of subspaces of Arnold–Douglas–Gupta [1], which were constructed by using the composite elements. The global mixed finite element approximation to (2.4) is defined by restricting (2.4) on the finite dimensional subspace $\underset{\approx}{H}^h \times \underset{\sim}{V}^h$. We remark that the mixed finite element subspaces $\underset{\approx}{H}^h \times \underset{\sim}{V}^h$ cited above were originally introduced for stationary elasticity problems which are coercive, it is necessary to show that we still can use these subspaces to approximate the noncoercive problem (2.4). We show this by using the duality argument due to Douglas and Roberts [9].

Since in each space $\underset{\sim}{V}^h$ in the family of mixed finite element spaces referenced above the vector functions $\underset{\sim}{v}_h \in \underset{\sim}{V}^h$ are allowed to be discontinuous across Γ_{jk}. As a consequence, attempting to impose the transmission condition (2.6) would include a flux conservation error; i.e., (2.5.ii) would not be satisfied unless the approximate solution $\underset{\sim}{v}_h \in \underset{\sim}{V}^h$ to the discrete analogue of (2.4) is a constant, a uninteresting case. Similar as in [8] this difficulty can be overcome by the hybridization; i.e., by introducing Lagrange multipliers ([4], [8]) $\{\lambda_{jk}\}$ on the interface $\{\Gamma_{jk}\}$.

Let $\underset{\approx}{H}^h \times \underset{\sim}{V}^h$ be either the Johnson–Mercier space or Arnold–Douglas–Gupta spaces. Let $P_k(\Gamma_{j\ell})$ denote the space of polynomials of degree less than or equal to k on $\Gamma_{j\ell}$. Set

$$\underset{\approx}{H}^h_j = \underset{\approx}{H}^h|_{\Omega_j}, \qquad \underset{\sim}{V}^h_j = \underset{\sim}{V}^h|_{\Omega_j}, \quad \underset{\sim}{M}^h_{j\ell} = \underset{\sim}{P}_k(\Gamma_{j\ell}) \quad \text{for } \Gamma_{j\ell} \neq \emptyset.$$

The (global) mixed finite element approximation to (2.4) is defined as follows: Seek $(\underset{\approx}{\sigma}_h, \underset{\sim}{u}_h) \in \underset{\approx}{H}^h \times \underset{\sim}{V}^h$ such that

$$\text{(3.1.i)} \qquad a(\underset{\approx}{\sigma}_h, \underset{\approx}{\tau}_h)_\Omega + (\underset{\sim}{u}_h, \operatorname{div} \underset{\approx}{\tau}_h)_\Omega - i\omega^{-1}\langle A^{-1}\underset{\approx}{\sigma}_h\underset{\sim}{\nu}, \underset{\approx}{\tau}_h\underset{\sim}{\nu}\rangle_\Gamma = 0 \quad \underset{\approx}{\tau}_h \in \underset{\approx}{H}^h,$$

$$\text{(3.1.ii)} \qquad (\operatorname{div} \underset{\approx}{\sigma}_h, \underset{\sim}{v}_h)_\Omega + \omega^2(\underset{\sim}{u}_h, \underset{\sim}{v}_h)_\Omega = -(\underset{\sim}{f}, \underset{\sim}{v}_h)_\Omega, \qquad \underset{\sim}{v}_h \in \underset{\sim}{V}^h.$$

Theorem 3.1 (cf. [11]). *There exists an $h_0 > 0$ such that, for all $h \in (0, h_0]$, Problem (3.1) has a unique solution $(\underset{\approx}{\sigma}_h, \underset{\sim}{u}_h) \in \underset{\approx}{H}^h \times \underset{\sim}{V}^h$. Moreover, suppose that $\underset{\sim}{u} \in \underset{\sim}{H}^2(\Omega)$, then*

$$\text{(3.2.i)} \qquad \|\underset{\approx}{\sigma} - \underset{\approx}{\sigma}_h\|_{0,\Omega} \leq C\|\underset{\sim}{u}\|_{2,\Omega}h,$$

$$\text{(3.2.ii)} \qquad \|\underset{\sim}{u} - \underset{\sim}{u}_h\|_{0,\Omega} \leq C\|\underset{\sim}{u}\|_{2,\Omega}h^{\min(2,k)},$$

for some positive constant C, which depends only on Lamé constants λ_r, λ_i, μ_r, μ_i, Ω and the frequency ω.

To define the discrete iterative algorithm analogous to (2.7), we notice that (3.1) has the following equivalent hybrid formulation: Seek $(\underset{\approx}{\sigma}_{hj}, \underset{\sim}{u}_{hj}, \underset{\sim}{\lambda}_{hjk}) \in \underset{\approx}{H}^h_j \times \underset{\sim}{V}^h_j \times \underset{\sim}{M}^h_{jk}$ such that for

$j, k = 1, 2, \ldots, J$,

(3.3.i) $$a(\underset{\approx}{\sigma}_{hj}, \tau_h)_{\Omega_j} + (\underset{\sim}{u}_{hj}, \operatorname{div} \underset{\approx}{\tau}_h)_{\Omega_j} - i\omega^{-1}\langle A^{-1}\underset{\approx}{\sigma}_{hj}\underset{\sim}{\nu}_j, \underset{\approx}{\tau}_h\underset{\sim}{\nu}_j\rangle_{\Gamma_j} - \sum_{k=1}^{J}\langle \underset{\sim}{\lambda}_{hjk}, \underset{\approx}{\tau}_{jk}\underset{\sim}{\nu}_j\rangle_{\Gamma_{jk}} = 0, \quad \underset{\approx}{\tau}_h \in \underset{\approx}{H}{}_j^h,$$

(3.3.ii) $$(\operatorname{div} \underset{\approx}{\sigma}_{hj}, \underset{\sim}{v}_h)_{\Omega_j} + \omega^2(\underset{\sim}{u}_{hj}, \underset{\sim}{v}_h)_{\Omega_j} = -(\underset{\sim}{f}, \underset{\sim}{v}_h)_{\Omega_j}, \quad \underset{\sim}{v}_h \in \underset{\sim}{V}{}_j^h,$$

(3.3.iii) $$\underset{\approx}{\sigma}_{hj}\underset{\sim}{\nu}_j + \alpha\underset{\sim}{\lambda}_{hjk} = -\underset{\approx}{\sigma}_{hk}\underset{\sim}{\nu}_k + \alpha\underset{\sim}{\lambda}_{hkj}, \quad \text{on } \Gamma_{jk}.$$

Base on the hybrid formulation (3.3), we define the domain decomposition iterative procedure analogous to (2.7) as follows: For all j and k, choose $\underset{\approx}{\sigma}{}_{hj}^0 \in \underset{\approx}{H}{}_j^h$, $\underset{\sim}{u}{}_{hj}^0 \in \underset{\sim}{V}{}_j^h$, $\underset{\sim}{\lambda}{}_{hjk}^0 \in \underset{\sim}{M}{}_{jk}^h$ arbitrarily, then compute $\{\underset{\approx}{\sigma}{}_{hj}^n, \underset{\sim}{u}{}_{hj}^n, \underset{\sim}{\lambda}{}_{hjk}^n\} \in \underset{\approx}{H}{}_j^h \times \underset{\sim}{V}{}_j^h \times \underset{\sim}{M}{}_{jk}^h$ recursively by solving the following equations:

(3.4.i) $$a(\underset{\approx}{\sigma}{}_{hj}^n, \underset{\approx}{\tau}_h)_{\Omega j} + (\underset{\sim}{u}{}_{hj}^n, \operatorname{div} \underset{\approx}{\tau}_h)_{\Omega_j} - i\omega^{-1}\langle A^{-1}\underset{\approx}{\sigma}{}_{hj}^n\underset{\sim}{\nu}_j, \underset{\approx}{\tau}_h\underset{\sim}{\nu}_j\rangle_{\Gamma_j} - \sum_{k=1}^{J}\langle \underset{\sim}{\lambda}{}_{hjk}^n, \underset{\approx}{\tau}_h\underset{\sim}{\nu}_j\rangle_{\Gamma_{jk}} = 0, \quad \underset{\approx}{\tau}_h \in \underset{\approx}{H}{}_j^h,$$

(3.4.ii) $$(\operatorname{div} \underset{\approx}{\sigma}{}_{hj}^n, \underset{\sim}{\nu}_h)_{\Omega_j} + \omega^2(\underset{\sim}{u}{}_{hj}^n, \underset{\sim}{\nu}_h)_{\Omega_j} = -(\underset{\sim}{f}, \underset{\sim}{\nu}_h)_{\Omega_j}, \quad \underset{\sim}{v}_h \in \underset{\sim}{V}{}_j^h,$$

(3.4.iii) $$\underset{\approx}{\sigma}{}_{hj}^n\underset{\sim}{\nu}_j + \alpha\underset{\sim}{\lambda}{}_{hjk}^n = -\underset{\approx}{\sigma}{}_{hk}^{n-1}\underset{\sim}{\nu}_k + \alpha\underset{\sim}{\lambda}{}_{hkj}^{n-1}, \quad \text{on } \Gamma_{jk}.$$

The main result of this section is the following two theorems, see [11] for the detailed exposition.

Theorem 3.2. *Choose α such that $\alpha_i\lambda_i - \alpha_r\lambda_r > 0$ $\alpha_i\mu_i - \alpha_r\mu_r > 0$, then for $h \le h_0$ the iterates $\{\underset{\approx}{\sigma}{}_{hj}^n, \underset{\sim}{u}{}_{hj}^n, \underset{\sim}{\lambda}{}_{hjk}^n\}$ defined above converges to the solution $\{\underset{\approx}{\sigma}_{hj}, \underset{\sim}{u}_{hj}, \underset{\sim}{\lambda}_{hjk}\}$ of the (global) hybridized mixed finite element procedure (4.3) in the following sense:*

$$\text{(i)}.\ \underset{\approx}{\sigma}{}_{hj}^n \longrightarrow \underset{\approx}{\sigma}_{hj} \equiv \underset{\approx}{\sigma}_h|_{\Omega_j} \quad \text{in } \underset{\approx}{L}{}^2(\Omega_j),$$

$$\text{(ii)}.\ \underset{\sim}{u}{}_{hj}^n \longrightarrow \underset{\sim}{u}_{hj} \equiv \underset{\sim}{u}_h|_{\Omega_j} \quad \text{in } \underset{\sim}{L}{}^2(\Omega_j),$$

$$\text{(iii)}.\ \underset{\sim}{\lambda}{}_{hjk}^n \text{ and } \underset{\sim}{\lambda}{}_{hjk}^n \longrightarrow \underset{\sim}{\lambda}_{hjk} \quad \text{in } \underset{\sim}{L}{}^2(\Gamma_{jk}).$$

Define

$$\underset{\approx}{\pi}{}_{hj}^n = \underset{\approx}{\sigma}_{hj} - \underset{\approx}{\sigma}{}_{hj}^n, \qquad \underset{\sim}{e}{}_{hj}^n = \underset{\sim}{u}_{hj} - \underset{\sim}{u}{}_{hj}^n, \qquad \underset{\sim}{\xi}{}_{hjk}^n = \underset{\sim}{\lambda}_{hjk} - \underset{\sim}{\lambda}{}_{hjk}^n,$$

$$\underset{\sim}{\xi}{}_{hkj}^n = \underset{\sim}{\lambda}_{hkj} - \underset{\sim}{\lambda}{}_{hkj}^n, \qquad E_h^n = \sum_{k=1}^{J}\sum_{j=1}^{J}\int_{\Gamma_{kj}} |\underset{\approx}{\pi}_{hk}\underset{\sim}{\nu}_k + \alpha\underset{\sim}{\xi}{}_{hkj}^n|^2.$$

Then we have

Theorem 3.3. *Choose α such that $\alpha_i\lambda_i > 2\alpha_r\lambda_r$ $\alpha_i\mu_i > 2\alpha_r\mu_r$, then there holds*

$$E_h^{n+1} \le (1 - Ch)E_h^n, \tag{3.5}$$

for some positive constant C which is independent of h.

It follows from Theorem 3.3 that the iterative algorithm (3.4) converges at a rate which has an upper bound of the form $1 - Ch$. Similar to the differential case, the proofs of the above theorems are based on a lemma which is an analogue of Lemma 2.1.

Lemma 3.1 (cf. [11]). *There holds the following identity:*

$$E_h^{n+1} = E_h^n - \sum_{k=1}^{J} \Big\{ 4\alpha_i [\operatorname{Im} a(\underset{\approx}{\pi}_{hk}^n, \underset{\approx}{\pi}_{hk}^n)_{\Omega_k} + \omega^{-1} \langle A^{-1} \underset{\approx}{\pi}_{hk}^n \underset{\sim}{\nu}_k, \underset{\approx}{\pi}_{hk}^n \underset{\sim}{\nu}_k \rangle_{\Gamma_k}] + 4\alpha_r [\omega^2 \| \underset{\sim}{e}_{hk}^n \|_{0,\Omega_k}^2 - \operatorname{Re} a(\underset{\approx}{\pi}_{hk}^n, \underset{\approx}{\pi}_{hk}^n)_{\Omega_k}] \Big\}. \tag{2.10}$$

References

1. D. N. Arnold, J. Douglas, Jr. and C. P. Gupta, *A family of higher order finite element methods for plane elasticity*, Numer. Math. **45** (1984), 1–22.
2. L. S. Bennethum and X. Feng, *A domain decomposition method for solving a Helmhotz–like problem in elasticity by Wilson nonconforming element*, R.A.I.R.O., Modélisation Math. Anal. Numér. (to appear).
3. J. H. Bramble, J. E. Pasciak, J. Wang and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition*, Math. Comp. **57** (1991), 1–21.
4. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer–Verlag, New York, 1991.
5. X.-C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Statist. Comput. **13** (1992), 243–258.
6. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North–Holland, Amsterdam, 1978.
7. B. Després, *Méthodes de Décomposition de Domaines pour les Problémes de Propagation Dóndes en Régime Harmonique*, Ph. D. Thesis, Université Paris IX Dauphine, UER Mathématiques de la Décision, 1991.
8. J. Douglas, Jr., P. J. S. Paes Leme, J. E. Roberts, 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).
9. J. Douglas, Jr. and J. Roberts, *Global estimates for mixed methods for second order elliptic equations*, Math. Comp. **1** (1982), 91–103.
10. G. Duvaut and J. L. Lions, *Inequalities in Mechanics and Physics*, Springer-Verlag, Berlin, 1976.
11. X. Feng, *An parallel iterative method for solving elastic and nearly elastic wave equations in the frequency domain*, preprint.
12. R. Glowinski and M. F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, in Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988, 144–172.
13. C. Johnson and B. Mercier, *Some equilibrium finite element methods for two–dimensional elasticity problems*, Numer. Math. **30** (1978), 103–116.
14. P. L. Lions, *On the Schwartz alternating method I, III*, in First and Third International Symposium on Domain Decomposition Method for Partial Differential Equations, SIAM, Philadelphia, 1988 and 1990.
15. C. L. Ravazzoli, J. Douglas, Jr., J. E. Santos, and D. Sheen, *On the solution of the equations of motion for nearly elastic solids in the frequency domain*, Technical Report, Center for Applied Mathematics, Purdue University, 1992.
16. O. B. Widlund, *Some Schwarz methods for symmetric and nonsymmetric elliptic problems*, in Fifth Conference on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan etc., eds., SIAM, Philadelphia, PA, 1992.
17. J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review **34** (1992), 581-613.

Representation of Discrete Steklov-Poincare Operator Arising in Domain Decomposition Methods in Wavelet Basis

A. Jemcov and M.D. Matovic

Centre for advanced Gas Combustion Technology
Department of Mechanical Engineering
Queen's University
Kingston, Ontario, Canada K7L 3N6

Abstract

This paper examines the sparse representation and preconditioning of a discrete Steklov-Poincare operator which arises in domain decomposition methods. A non-overlapping domain decomposition method is applied to a second order self-adjoint elliptic operator (Poisson equation), with homogeneous boundary conditions, as a model problem. It is shown that the discrete Steklov-Poincare operator allows sparse representation with a bounded condition number in wavelet basis if the transformation is followed by thresholding and rescaling. These two steps combined enable the effective use of Krylov subspace methods as an iterative solution procedure for the system of linear equations.

Finding the solution of an interface problem in domain decomposition methods, known as a Schur complement problem, has been shown to be equivalent to the discrete form of Steklov-Poincare operator. A common way to obtain Schur complement matrix is by ordering the matrix of discrete differential operator in subdomain node groups then block eliminating interface nodes. The result is a dense matrix which corresponds to the interface problem. This is equivalent to reducing the original problem to several smaller differential problems and one boundary integral equation problem for the subdomain interface.

From the pseudodifferential point of view Steklov-Poincare operator could be regarded as an elliptic pseudodifferential operator of order one. More precisely, if we consider Calderon-Seeley projector for Poisson equation the Steklov-Poincare operator transforms Dirichlet into Neumann data acting as a Fredholm operator between Sobolev spaces $S : H^t(\Gamma) \rightarrow H^{t-1}(\Gamma)$. Therefore, Steklov-Poincare operator is an integral operator with non-local support of its Schwartz kernel.

One major characteristics of the non-local operators is that their finite dimensional representations result in dense matrices whose condition number is of the order $1/h$ for elliptic pseudodifferential operators. This is better than the condition number of matrices produced by the discretization of differential operators, which is of the order $1/h^2$ for the same class of differential operators. However, the advantage of better matrix conditioning is offset by the matrix fill-in rendering the use of iterative procedures ineffective.

The remedy proposed in this paper is to apply a discrete wavelet transform on the Schur complement matrix. The transformed matrix is almost sparse in a sense that the vast majority of entries are close to zero for large N (fine mesh). The thresholding procedure is applied to initially dense matrix reducing the number of non-zero entries to an order of $O(N \log N)$. Moreover, if the initial differential operator is strongly elliptic, diagonal rescaling in conjunction with thresholding efficiently bounds the transformed matrix

condition number. Therefore, the discrete Steklov-Poincare operator can be made sparse in wavelet space and its condition number significantly reduced by rescaling and thresholding combined. The condition number is mesh size and problem independent, i.e. thresholding and rescaling together make a nearly optimal preconditioner. One obvious advantage is that the preconditioner is determined a priori and its coefficients are problem independent. Another advantage is in easy parallelization of the algorithm since only matrix-matrix and matrix-vector multiplications are involved.

One restriction on this method is that the linear dimension of the matrix must be a power of two. Although this analysis is strictly true only for elliptic pseudodifferential operators it is expected that good results can be obtained in other cases involving the interface problem for differential operators due to their non-local nature.

Simplified Approaches to Some Nonoverlapping Domain Decomposition Methods

Jinchao Xu
Penn State University
(xu@math.psu.edu)

An attemp will be made in this talk to present various domain decomposition methods in a way that is intuitively clear and technically coherent and concise. The basic framework used for analysis is the "parallel subspace correction" or "additive Schwarz" method, and other simple technical tools include "local-global" and "global-local" techniques, the formal one is for constructing subspace preconditioner based on a preconditioner on the whole space whereas the later one for constructing preconditioner on the whole space based on a subspace preconditioner.

The domain decomposition methods discussed in this talk fall into two major categories: one, based on local Dirichlet problems, is related to the "substructuring method" and the other, based on local Neumann problems, is related to the "Neumann-Neumann method" and "balancing method". All these methods will be presented in a systematic and coherent manner and the analysis for both two and three dimensional cases are carried out simultaneously. In particular, some intimate relationship between these algorithms are observed and some new variants of the algorithms are obtained.

This talk is based on a joint paper with Jun Zou.

Topic: *Session Chair:* *Room B*
Krylov Methods *Anne Greenbaum*

10:30 - 11:00	F. Campos	The Adaptive CCCG(n) Method for Efficient Solution of Time Dependent Partial Differential Equations
11:00 - 11:30	T. Barth	Conjugate Gradient Algorithms Using Multiple Recursions
11:30 - 12:00	J. Cullum	Iterative Methods for Solving Ax=b, GMRES/FOM versus QMR/BiCG

The adaptive CCCG(η) method for efficient solution of time dependent partial differential equations

Frederico F. Campos, filho
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
CP 702
30.161.970 Belo Horizonte - MG
Brazil
email ffcampos@dcc.ufmg.br

Nick R. C. Birkett
Oxford University Computing Laboratory
Numerical Analysis Group
Wolfson Building
Parks Road
Oxford, England OX1 3QD
email nrcb@comlab.oxford.ac.uk

December 22, 1995

Summary
The Controlled Cholesky factorisation has been shown to be a robust preconditioner for the Conjugate Gradient method. In this scheme the amount of fill-in is defined in terms of a parameter η, the number of extra elements allowed per column. It is demonstrated how an optimum value of η can be automatically determined when solving time dependent p.d.e.'s using an implicit time step method. A comparison between CCCG(η) and the standard ICCG solving parabolic problems on general grids shows CCCG(η) to be an efficient general purpose solver.

1 Introduction

A common problem in solving time dependent partial differential equations, using an implicit time-stepping method, is finding an efficient linear solver, given that a large number of algebraic systems have to be solved. If the algebraic systems are symmetric, real, and positive definite, then one popular choice of solver is the Incomplete Cholesky Conjugate Gradient method (ICCG) due to Meijerink and van der Vorst [7]. However, for general problems, particularly on unstructured meshes, this method may not be robust. Attempts to improve on the Incomplete Cholesky preconditioner have led to incomplete Cholesky with levels of fill-in and the use of drop tolerances schemes [8] to improve the approximation of the system by the preconditioner. However, both of these latter methods lead to preconditioners with unpredictable storage requirements.

Recently a new method of forming an approximate factorisation has been given by Campos and Rollett [3]. This new factorisation is both robust, in that it leads to good preconditioners, and predictable in its storage requirements. Furthermore, for a given amount of storage, the method can be coupled to a cheap optimisation process in order to find the best preconditioner. In a time-stepping procedure, where the number of timesteps to be taken is relatively large, the first few timesteps may be used to set up an optimal preconditioner. It is shown that for a range of test problems the increase in efficiency of using this approach can be quite dramatic.

2 The CCF preconditioner

The controlled Cholesky factorisation (CCF) [3, 2] is based on the minimisation of the Frobenius norm of $E = L - \tilde{L}$, where L is the factor obtained when the factorisation of a matrix A of order n is complete and $\tilde{L}$ when it is incomplete :

$$\text{minimise } \|E\|_F^2 = \sum_{j=1}^{n} c_j \text{ with } c_j = \sum_{i=1}^{n} |l_{ij} - \tilde{l}_{ij}|^2.$$

If c_j is split in two summations

$$c_j = \sum_{k=1}^{m_j+\eta} |l_{i_k j} - \tilde{l}_{i_k j}|^2 + \sum_{k=m_j+\eta+1}^{n} |l_{i_k j}|^2,$$

where m_j is the number of non-zero elements below the diagonal in the jth column of matrix A and η is the number of extra elements allowed per column. The first summation contains all $m_j + \eta$ non-zero elements of the jth column of $\tilde{L}$, and the second summation has only those remaining elements of the complete factor L which do not have corresponding elements in $\tilde{L}$. Considering that $\tilde{l}_{ij} \approx l_{ij}$ and l_{ij} is not computed, $\|E\|_F$ is minimised

based on a heuristic, which consists of modifying the first summation. By increasing η, allowing more fill-in, c_j will decrease simply because the first summation contains more terms and the second less. This is the same non-selective minimisation that occurs when levels of fill-in are used. Moreover, $\|E\|_F$ is further minimised by choosing the $m_j + \eta$ **largest** of $\tilde{L}$ to almost annihilate the corresponding largest elements in L leaving only the smallest l_{ij} in the second summation. This is a selective minimisation similar to a drop tolerance scheme [8].

The CCF algorithm also employs the modified Cholesky factorisation (MCF) of Gill and Murray [5, 6] to avoid loss of positive definiteness and increase robustness. In incomplete Cholesky decomposition (ICD) fill-in is determined by adding levels resulting in uncontrolled storage demand; the storage required by CCF, in comparison, is predictable as shown in Table 1. A comparison of CCCG(η), $\eta = 5$, 10, 20 with the MA31 subroutine

matrices	maximum bytes required	for i=4 and r=8
A	$(i+r)m+in+i$	$12m+4n+4$
$A+\tilde{L}_{IC(0)}$	$(i+2r)m+in+i$	$20m+4n+4$
$A+\tilde{L}_{CC(\eta)}$	$(3i+2r)m+((2i+r)\eta+2i)n+2i$	$28m+(16\eta+8)n+8$

Table 1: Comparison between storage demands of ICD(0) and CCF(η).
n : order of A, m : number of non-zero elements of A (excluding those above diagonal), η : number of extra elements per column, i, r : number of bytes for integer and real variables, respectively.

using drop tolerance $C = 10^{-1}$, 10^{-2}, 10^{-4} solving ten systems from the Release I of the Harwell-Boeing sparse matrix collection [4] was made by Campos [2]. MA31 is from the Harwell Subroutine Library and is an Incomplete Cholesky Conjugate Gradient method with a drop tolerance scheme proposed by Munksgaard [8]. A selection of results are presented in Table 2.

These results reveal that for all test cases where the drop tolerance scheme failed, CCCG(η) succeeded using less storage.

3 An adaptive preconditioner for CG

When solving systems of equations with multiple right hand sides it is possible to estimate the optimum η_{opt} for which the total time t is minimum, where

$$t = mp + si, \tag{1}$$

m is the number of systems required to be solved to find η_{opt}, s is the total number of systems to be solved and p and i are the preconditioning and iteration time, respectively.

System bcsstk	iterations C for MA31(C)			number of non-zeros C for MA31(C)			total time (seconds) C for MA31(C)		
	10^{-1}	10^{-2}	10^{-4}	10^{-1}	10^{-2}	10^{-4}	10^{-1}	10^{-2}	10^{-4}
08	85	26	8	13026	13953	20795	3.27	1.46	1.76
10	nc	nc	10	22477	26141	35650	—	—	2.81
14	nc	93	8	65324	69347	102615	—	16.23	11.09
16	ow	62	13	295081	307333	678790	—	56.56	195.76
System bcsstk	iterations η for CCCG(η)			number of non-zeros η for CCCG(η)			total time (seconds) η for CCCG(η)		
	5	10	20	5	10	20	5	10	20
08	14	12	11	12143	17316	27533	2.34	3.14	5.07
10	32	10	1	16914	20350	20771	2.62	1.36	0.76
14	25	19	12	41311	49922	66688	6.67	6.96	8.17
16	21	18	14	171615	195569	243317	33.12	35.40	41.99

Table 2: Comparison among MA31(C) and CCCG(η).
nc : process does not converge with tolerance $\|r_k\|/\|r_0\| \leq 10^{-10}$ or min(Order,1000) iterations and ow : overflow during factorisation.

Typically, $m \approx 10$ and if $s \approx m$ in (1) then $t = s(p+i)$ needs to be minimised. On the other hand, if $s \gg m$ then it is $t = si$ that has to be minimised. Then a crude approximation to η_{opt} may be found using, for instance, the Brent method [1] and some system calls to the elapsed cpu time.

This technique has been successfully applied to the solution of parabolic problems using implicit timestepping methods.

4 Numerical example

In order to demonstrate the efficacy of the CCCG algorithm, consider the model problem :

$$\frac{\partial u}{\partial t} - \nabla . D \nabla u = f(x,y,t), \text{ for } (x,y) \in \Omega \subset \mathbb{R}^2, \text{ and } t \in (0,T]. \quad (2)$$

Here $D(x,y)$ is a 2×2 symmetric positive definite diffusion matrix, and $f(x,y,t)$ is a known forcing function. For simplicity zero boundary conditions were imposed on all boundaries. Since the only coefficient in the problem which varies with t is f, the differential operator is time independent.

This problem is discretized in space using standard Galerkin Finite Elements on linear triangles [11], and discretized in time using the backward Euler method for the differential

part, and Trapezoidal quadrature for the coefficient f, leading to the implicit time stepping formula :

$$(M + \Delta t K)U^{n+1} = MU^n + \frac{\Delta t}{2}(F^n + F^{n+1}), \tag{3}$$

where M is the positive definite mass matrix and K the positive semi-definite stiffness matrix, and F^n load vectors. This discrete problem is for illustrative purposes and other discretizations may be considered which lead to an implicit time-stepping scheme.

Assuming that Δt remains fixed, then it is required to solve (3) at each timestep, and it is to this sequence of linear algebraic systems that the CCCG algorithm will be applied. In the following problem, the solution times and best value η_{opt} of η, the number of extra elements per row of the preconditioner, are given, together with a comparison with the standard ICCG method. In each case the linear solver was required to reduce the initial residual by a factor of 10^6.

For the following problem the times indicated are the total times in seconds, on an IBM RS6000 model 550, to integrate the discrete system from $t = 0$ to $t = T$, including the calculation of the right hand sides, setting up the preconditioner and solution of all the linear systems. The maximum allowable value for η was set $\eta_{max} = 40$.

Anisotropic diffusion on a highly stretched mesh.

In this problem the solution region Ω and a coarse Navier-Stokes aerofoil mesh illustarted in Figure 1. A sequence of discretizations was obtained by uniform refinement of the grid shown. These grids have a maximum cell aspect ratio of 289, and therefore the condition of the resulting systems is likely to be very large [10]. The following data were used :

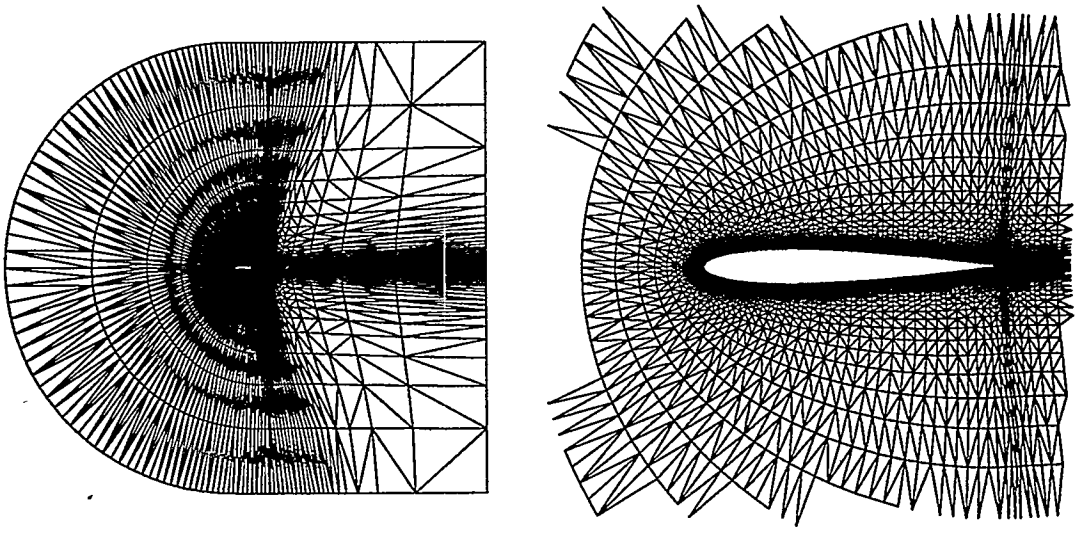
$$\begin{aligned}
u_0(x,y) &= 50\sin(\pi x)\,\sin(\pi y) \\
D &= \begin{pmatrix} 1.000 & 0.000 \\ 0.000 & 1.000E+04 \end{pmatrix}, \\
f(x,y,t) &= (-50\pi\sin(\pi t) + 100\pi^2\cos(\pi t))\sin(\pi x)\sin(\pi y), \\
T &= 20, \\
\Delta t &= \frac{T}{200}.
\end{aligned}$$

Convergence results are shown in Table 3.

Table 3 shows a dramatic difference between the performance of ICCG and CCCG, where the performance of the ICCG iteration degrades significantly as the mesh is refined. For

Number of unknowns	CCCG			ICCG	
	η_{opt}	cpu time	Iters/solve	cpu time	Iters/solve
4236	15	166.2	9	352.8	64
16664	40	805.1	6	11350.2	639
66096	40	5596.9	16	failed to converge	

Table 3: cpu seconds and iterations per solve for Problem 4.1.



(a) (b)

Figure 1: (a) Coarse grid (b) Detail.

other numerical test problems, not shown here, such as isotropic diffusion on uniform and and non-uniform meshes, and diffusion with discontinuous coefficients, the CCCG iteration proved significantly more robust and efficient than ICCG.

5 Conclusions

We have shown that, for implicit timestepping methods, a procedure based on the Controlled Cholesky Factorisation leads to a good preconditioner for the Conjugate Gradient method. Furthermore the method has predictable storage requirements, unlike techniques based on levels of fill-in or drop tolerances. Because the storage requirements of the Controlled Cholesky Factorisation are predictable, it is always possible to select an η_{max} within the limits of any machine.

The most attractive feature of this method is that the required preconditioner is arrived at automatically, removing, to a great extent, user intervention and thus making it of practical use as a general purpose iterative solver when coupled to CG.

References

[1] R. P. Brent, 1973. *Algorithms for Minimization Without Derivatives.* Prentice-Hall.

[2] F. F. Campos, filho, 1995. *Analysis of Conjugate Gradients - type methods for solving linear equations.* PhD thesis, Oxford University Computing Laboratory.

[3] F. F. Campos, filho and J. S. Rollett, March 1995. Controlled Cholesky Factorisation for preconditioning the Conjugate Gradient method. OUCL Report NA95/05, Oxford University Computing Laboratory, Oxford.

[4] I. S. Duff, R. G. Grimes and J. G Lewis, 1992. Users' guide for the Harwell-Boeing sparse matrix collection (release I). CERFACS TR/PA/92/86, CERFACS, ftp orion.cerfacs.fr.

[5] P. E. Gill and W. Murray, 1974. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Prog.*, **7**(3): 311–350.

[6] P. E. Gill, W. Murray and M. H. Wright, 1981. *Practical Optimization.* Academic Press.

[7] J. A. Meijerink and H. A. van der Vorst, 1977. An iterative solution method for linear system of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, **31**(137): 148–162.

[8] N. Munksgaard, 1980. Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients. *ACM Trans. Math. Software*, **6**(2): 206–219.

[9] L. E. Scales, 1985. *Introduction to Non-Linear Optimization.* MacMillan.

[10] G. Strang and G. Fix, 1973. *An analysis of the Finite Element Method.* Prentice Hall.

[11] O. C. Zienkiewicz, 1971. *The Finite Element Method in Engineering.* McGraw-Hill.

Conjugate Gradient Algorithms Using Multiple Recursions

Teri Barth and Tom Manteuffel[1]
University of Colorado

Much is already known about when a conjugate gradient method can be implemented with short recursions for the direction vectors. The work done in 1984 by Faber and Manteuffel [1] gave necessary and sufficient conditions on the iteration matrix A, in order for a conjugate gradient method to be implemented with a single recursion of a certain form. However, this form does not take into account all possible recursions. This became evident when Jagels and Reichel [3, 4] used an algorithm of Gragg for unitary matrices [2] to demonstrate that the class of matrices for which a practical conjugate gradient algorithm exists can be extended to include unitary and shifted unitary matrices. The implementation uses short double recursions for the direction vectors. This motivates the study of multiple recursion algorithms.

In this talk, we show that the conjugate gradient method for unitary and shifted unitary matrices can be implemented using a single short term recursion of a special type called an (ℓ, m) recursion with $\ell, m \leq 1$. We then examine the class of matrices for which a conjugate gradient method can be carried out using a general (ℓ, m) recursion. This class includes the class of normal matrices with rational degree (ℓ, m) as well as low rank perturbations of these matrices.

Under some circumstances, an (ℓ, m) recursion can break down. We also show that any (ℓ, m) recursion can be reformulated as as m short recursions that will not break down.

References

[1] V. Faber and T. A. Manteuffel, *Necessary and Sufficient Conditions for the Existence of a Conjugate Gradient Method*, SIAM J. Numer. Analysis, Vol. 21, No. 2, (1984) pp. 352-362.

[1]Speaker

[2] W. B. Gragg, *Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle*, J. Comp. Appl. Math, 46 (1993), pp. 183-198.

[3] C. F. Jagels, and L. Reichel, *The isometric Arnoldi process and an application to iterative solution of large linear systems*, in Iterative Methods in Linear Algebra, eds. R. Beauwens and P. de Groen, Elsevier, Amsterdam, 1992, pp. 361-369.

[4] C. F. Jagels, and L. Reichel, *A Fast Minimal Residual Algorithm For Shifted Unitary Matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 555-570.

Iterative methods for solving Ax=b, GMRES/FOM versus QMR/BiCG

Jane Cullum
Mathematical Sciences Department,
IBM Research Division,
T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA.
E-mail: cullumj@watson.ibm.com

This work was supported by
NSF grant GER-9450081.

We study the convergence of GMRES/FOM and QMR/BiCG methods for solving nonsymmetric Ax=b. We prove that given the results of a BiCG computation on Ax=b, we can obtain a matrix B with the same eigenvalues as A and a vector c such that the residual norms generated by a FOM computation on Bx=c are identical to those generated by the BiCG computations. Using a unitary equivalence for each of these methods, we obtain test problems where we can easily vary certain spectral properties of the matrices. We use these test problems to study the effects of nonnormality on the convergence of GMRES and QMR, to study the effects of eigenvalue outliers on the convergence of QMR, and to compare the convergence of restarted GMRES, QMR, and BiCGSTAB across a family of normal and nonnormal problems. Our GMRES tests on nonnormal test matrices indicate that nonnormality can have unexpected effects upon the residual norm convergence, giving misleading indications of superior convergence over QMR when the error norms for GMRES are not significantly different from those for QMR. Our QMR tests indicate that the convergence of the QMR residual and error norms is influenced predominantly by small and large eigenvalue outliers and by the character, real, complex, or nearly real, of the outliers and the other eigenvalues. In our comparison tests QMR outperformed GMRES(10) and GMRES(20) on both the normal and nonnormal test matrices.

Topic: *Domain Decom*	*Session Chair:* *Olof Widlund*	*Room A*
4:45 - 5:15	S. Maliassov	Domain Decomposition Method for Nonconforming Finite Element Approximations of Anisotropic Elliptic Problems on Nonmatching Grids
5:15 - 5:45	H. Zhao	Analysis of Generalized Schwarz Alternating Procedure for Domain Decomposition
5:45 - 6:15	R. Tezaur	Substructuring by Lagrange Multipliers for Solids and Plates
6:15 - 6:45	X.C. Tai	Some Nonlinear Space Decomposition Algorithms

DOMAIN DECOMPOSITION METHOD FOR NONCONFORMING FINITE ELEMENT APPROXIMATIONS OF ANISOTROPIC ELLIPTIC PROBLEMS ON NONMATCHING GRIDS

S.Y. MALIASSOV*

Abstract

An approach to the construction of an iterative method for solving systems of linear algebraic equations arising from nonconforming finite element discretizations with nonmatching grids for second order elliptic boundary value problems with anisotropic coefficients is considered. The technique suggested is based on decomposition of the original domain into nonoverlapping subdomains. The elliptic problem is presented in the macro-hybrid form with Lagrange multipliers at the interfaces between subdomains. A block diagonal preconditioner is proposed which is spectrally equivalent to the original saddle point matrix and has the optimal order of arithmetical complexity. The preconditioner includes blocks for preconditioning subdomain and interface problems. It is shown that constants of spectral equivalence are independent of values of coefficients and mesh step size.

* S.Yu. Maliassov, Institute for Scientific Computation and Department of Mathematics, Texas A&M University, 505 Blocker Bldg., College Station, TX 77843–3404, U.S.A.

ANALYSIS OF GENERALIZED SCHWARZ ALTERNATING PROCEDURE FOR DOMAIN DECOMPOSITION

BJORN ENGQUIST * AND HONGKAI ZHAO †

Abstract.

The Schwartz alternating method(SAM) is the theoretical basis for domain decomposition which itself is a powerful tool both for parallel computation and for computing in complicated domains. The convergence rate of the classical SAM is very sensitive to the overlapping size between each subdomain, which is not desirable for most applications. We propose a generalized SAM procedure which is an extension of the modified SAM proposed by P.-L. Lions in [4]. Instead of using only Dirichlet data at the artificial boundary between subdomains, we take a convex combinaton of u and $\frac{\partial u}{\partial n}$, i.e. $\frac{\partial u}{\partial n} + \Lambda u$, where Λ is some "positive" operator. Convergence of the modified SAM without overlapping in a quite general setting has been proven by P.-L.Lions using delicate energy estimates. The important questions remain for the generalized SAM. (1)What is the most essential mechanism for convergence without overlapping? (2)Given the partial differential equation, what is the best choice for the positive operator Λ ? (3)In the overlapping case, is the generalized SAM superior to the classical SAM? (4)What is the convergence rate and what does it depend on? (5)Numerically can we obtain an easy to implement operator Λ such that the convergence is independent of the mesh size. All these questions are addressed in this paper.

To analyze the convergence of the generalized SAM we focus, for simplicity, on the Possion equation for two typical geometry in two subdomain case. From the analysis we can see clearly that the generalized SAM converges for the following two reasons

(i) Maximum principle or variational interpretation(iterated projections) if there is overlap, which is also true for the classical SAM. ([2], [3])

(ii) "Positivity" of the Dirichlet to Neuman operator which gives convergence even with no overlap. This makes the generalized SAM better than the classical SAM.

In the most interesting case where the two subdomains are of a comparable size which is much larger than the overlapping size, the convergence rate r_G is asympototically

$$r_G \approx \left|(D_1 + \Lambda_1)^{-1}(D_2 - \Lambda_1)(D_2 + \Lambda_2)^{-1}(D_1 - \Lambda_2)\right|_{H^1} e^{-\delta} \tag{1}$$

Each D_i is the Dirichlet to Neuman operator in Ω_i and δ is the size of overlap.

For more general elliptic operators in more complicated geometries with two subdomains, we can use the equivalence of elliptic operators and a transformation which reduces them to the two previous cases. We can also extend this generalized SAM method to the multidomain case by reducing it to the two subdomain case. Motivated by [1] we use some local operators which might involve the tangential information to approximate the Dirichlet to Neuman operator. These numerical schemes can be easily incooperated into the existing code for domain decomposition to improve the performance since they do not change the data structure in the interior of the domain or subdomain.

* Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095-1555, <engquist@math.ucla.edu>

† Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095-1555, <hzhao@math.ucla.edu>

Substructuring by Lagrange multipliers for solids and plates

Jan Mandel and Radek Tezaur
Center for Computational Mathematics
University of Colorado at Denver
Denver CO 80217-3364

Charbel Farhat
Center for Aerospace Structures
University of Colorado at Boulder
Boulder, CO 80309-0429

We present principles and theorereticaI foundation of a substructuring method for large structural problems. The algorithm is preconditioned conjugate gradients on a subspace for the dual problem. The preconditioning is proved asymptotically optimal and the method is shown to be parallel scalable, i.e., the condition number is bounded independently of the number of substructures.

For plate problems, a special modification is needed that retains continuity of the displacement solution at substructure crosspoints, resulting in an asymptically optimal method.

The results are confirmed by numerical experiments.

SOME NONLINEAR SPACE DECOMPOSITION ALGORITHMS

Xue–Cheng Tai and Magne Espedal

Department of Mathematics,
University of Bergen,
Alleg. 55, 5007,
Bergen, Norway.

ABSTRACT. Convergence of a space decomposition method is proved for a general convex programming problem. The space decomposition refers to methods that decompose a space into sums of subspaces, which could be a domain decomposition or a multigrid method for partial differential equations. Two algorithms are proposed. Both can be used for linear as well as nonlinear elliptic problems and they reduce to the standard additive and multiplicative Schwarz methods for linear elliptic problems. Two "hybrid" algorithms are also presented. They converge faster than the additive one and have better parallelism than the multiplicative method. Numerical tests with a two level domain decomposition for linear, nonlinear and interface elliptic problems are presented for the proposed algorithms.

CONTENTS

1. INTRODUCTION

This work presents a general space decomposition method for convex programming problems and gives an estimation of the rate of convergence of the method. One intension is to use the method to solve linear and nonlinear elliptic partial differential equations by domain decomposition or multilevel methods. In the applications given in [24], a two level overlapping domain decomposition method is considered.

The essence of the proposed method is to decompose the minimization space into a sum of subspaces and then solve the original minimization problem sequentially or in parallel over each of the subspaces. Due to the fact that the decomposed spaces can be arbitrary, especially since they are not orthogonal to each other, the usual convergence proofs for block relaxation methods cannot be used here to predict the convergence. However, using the experiences from domain decomposition and multigrid methods, we assume that the decomposed spaces satisfy a certain "spectral" bound, see constants C_1 and C_2 in (2.8) and (2.9), and then use these constants to estimate the convergence rate of the proposed methods.

The proposed algorithms are given for a convex programming problem. We expect that they could also be used to get efficient algorithms for some optimal control problems related to partial differential equations, see Kunisch and Tai [18] and [19] for applications.

The two level domain decomposition method can be viewed a space decomposition is inspired by the work of Xu [29], where it was observed that domain decomposition methods, multilevel methods and multigrid methods can be viewed in some way as space decomposition techniques and many of the methods proposed in literature for the above mention techniques are in essence similar to the Gauss-Seidel or Jacobi method. In Smith, Bjorstad and Gropp [28], by following the works of [13], [14], [32] and [10], etc. an abstract convergence was given for linear self-adjoint

1991 *Mathematics Subject Classification.* 65J10, 65M55, 65Y05.

Key words and phrases. Parallel, domain decomposition, nonlinear, elliptic equation, space decomposition.

The work was supported by by VISTA, a research cooperation between the Norwegian Academy of Science and Letters and Den norske oljeselskap a.s. (Statoil).

and also indefinite and nonsymmetric problems. Two schemes are proposed in this work. They could be used both for linear and nonlinear elliptic problems. In the linear case, they reduce to the standard additive and multiplicative Schwarz methods. Therefore, the algorithms generalise the known additive and multiplicative methods to certain nonlinear cases. Due to appearance of the nonlinearity, a modified abstract convergence theory is given.

The well-known substructuring BPS (see [5], [6], [7]) and BEPS (see [4]) preconditioners use nonoverlapping subdomains, see also [3], [21]. For a nonoverlapping domain decomposition, a finite element function w can be decomposed as $w = w_p + w_H$, here w_p has zero trace on the interfaces and equals to w in the interior nodes of the substructures and w_H equals to w on the interfaces and is extended to the interior by harmonic extension. If we use Gauss-Seidel iteration, we get the exact solution in one iteration. However, to get the harmonic extension w_H is equivalent to solving the original problem. The construction of the preconditioners in [6]–[7] and [4] can be regarded as Jacobi iteration with approximate solvers for the harmonic extensions. The methods of [5] and [21] is a Gauss-Seidel iteration with a further suitable decomposition for w_H. By using a slightly different decomposition, in Espedal and Ewing [16, p. 125], a parallel nonoverlapping method was derived for solving a linearised two-phase immiscible flow. We hope that by viewing the construction of nonoverlapping preconditioners as an iterative approximate solving of a space decomposition, an abstract convergence analysis can also be obtained for them for some nonlinear problems.

In the literature, domain decomposition methods, multigrid methods and multilevel methods have been successfully used for different kinds of linear partial differential equations, see [17] and [22], [28], [29]. However, the results for using them for nonlinear problems are not as rich as for linear problems. In Cai and Dryja [9], a semilinear elliptic equation is first linearised by the Newton's method and then solved by the additive Schwarz scheme. In papers by Xu [30], [31], a two level method without doing domain decomposition is used for nonlinear elliptic problems. In Axellsson and Kaporin [1], a minimum residual adaptive multilevel method is given for some nonlinear problems. In Dawson and Wheeler [12], a two level method is used for a nonlinear parabolic equation; The work of Lions [20] seems to be the pioneering work for using domain decomposition methods for nonlinear partial differential equations. In Rannacher [23], a Newton type algorithm is studied for nonlinear elliptic problems. Multigrid methods for nonlinear problems are studied by Bank [2], Brandt [8], etc. For some earlier works of the authors related to this one, consult [24] and [25]–[27].

When we apply the methods here for a nonlinear problem, we need to solve many smaller size problems in an iterative way and this iterative procedure convergence as "quickly" as for linear problems. For some nonlinear problems, by reducing the large size problem into many smaller size problems and then linearising the smaller size problems, substantial computational efforts can be saved compared to first linearising and then decomposing the problem.

2. Statement of the problem and the algorithms

Consider the nonlinear problem

$$\min_{v \in V} F(v) \ . \tag{2.1}$$

Above, the function F is differentiable and convex, the space V is a reflexive Banach space. One knows that partial differential equations of the type

$$-\sum D_i(a_{ij} D_j u) + bu = f \text{ in } \Omega \ ,$$

and

$$-\nabla \cdot (\rho(|\nabla u|)\nabla u) = f \text{ in } \Omega \ ,$$

with a suitably given ρ, can be solved by (2.1) by defining the function F and space V properly.

We shall use space decomposition methods to solve (2.1). A space decomposition method refers to a method that decomposes the space V into a sum of subspaces, i.e. there are spaces V_i, $i = 1, 2, \cdots, m$ such that

$$V = V_1 + V_2 + \cdots + V_m \ . \tag{2.2}$$

The meaning of the above decomposition is that $\forall v$, there exists $v_i \in V_i$ such that $v = \sum_{i=1}^m v_i$ and on the other hand, if $v_i \in V_i$, then $\sum_{i=1}^m v_i \in V$. If the space can be decomposed as in (2.2), then the followings algorithms can be used to solve (2.1).

Algorithm 2.1. (An additive space decomposition method).

Step 1. *Choose initial values* $u_i^0 = u^0 \in V$ *and relaxation parameters* $\alpha_i > 0$ *such that* $\sum_{i=1}^m \alpha_i \leq 1$.

Step 2. *For* $n \geq 0$, *find* $u_i^{n+\frac{1}{2}} \in V_i$ *in parallel for* $i = 1, 2, \cdots, m$ *such that*

$$F\left(\sum_{k=1,k\neq i}^{m} u_k^n + u_i^{n+\frac{1}{2}}\right) \leq F\left(\sum_{k=1,k\neq i}^{m} u_k^n + v_i\right) , \quad \forall v_i \in V_i . \tag{2.3}$$

Step 3. *Set*

$$u_i^{n+1} = u_i^n + \alpha_i(u_i^{n+\frac{1}{2}} - u_i^n) , \tag{2.4}$$

and go to the next iteration.

Algorithm 2.2. (A multiplicative space decomposition method).

Step 1. *Choose initial values* $u_i^0 = u^0 \in V$.

Step 2. *For* $n \geq 0$, *find* $u_i^{n+1} \in V_i$ *sequentially for* $i = 1, 2, \cdots, m$ *such that*

$$\begin{aligned} &F\left(\sum_{1\leq k<i} u_k^{n+1} + u_i^{n+1} + \sum_{i<k\leq m} u_k^n\right) \\ \leq &F\left(\sum_{1\leq k<i} u_k^{n+1} + v_i + \sum_{i<k\leq m} u_k^n\right) , \quad \forall v_i \in V_i . \end{aligned} \tag{2.5}$$

Step 3. *Go to the next iteration.*

In the following, the notation $\langle \cdot, \cdot \rangle$ is used to denote the duality pairing between V and V'. Function F is assumed to be Gateaux differentiable (see [11]) and there are constants $K > 0$, $L < \infty$ such that

$$\begin{aligned} \langle F'(w) - F'(v), w - v\rangle &\geq K\|w - v\|_V^2 , \quad \forall w, v \in V , \\ \|F'(w) - F'(v)\|_{V'} &\leq L\|w - v\|_V , \quad \forall w, v \in V , \end{aligned} \tag{2.6}$$

and from which, it is easy to deduce that

$$K\|w - v\|_V^2 \leq \langle F'(w) - F'(v), w - v\rangle \leq L\|w - v\|_V^2 , \quad \forall w, v \in V . \tag{2.7}$$

Under assumption (2.6), problem (2.1) and subproblems (2.5) and (2.3) have unique solutions, see [15, p. 35].

For the decomposed spaces, we assume that there is a constant $C_1 > 0$ such that $\forall v \in V$, we can find $v_i \in V_i$ to satisfy:

$$v = \sum_{i=1}^{m} v_i , \ \textit{and} \qquad \sum_{i=1}^{m} \|v_i\|_V^2 \leq C_1^2\|v\|_V^2 . \tag{2.8}$$

Moreover, assume that there is a $C_2 > 0$ such that there holds

$$\begin{aligned} &\sum_{i=1}^{m}\sum_{j=1}^{m}\langle F''(w_{ij})u_i, v_j\rangle \leq C_2\left(\sum_{i=1}^{m}\|u_i\|_V^2\right)^{\frac{1}{2}}\left(\sum_{i=1}^{m}\|v_i\|_V^2\right)^{\frac{1}{2}} , \\ &\forall w_{ij} \in V, \forall u_i \in V_i, \forall v_j \in V_j . \end{aligned} \tag{2.9}$$

Domain decomposition methods, multilevel methods and multigrid methods can be viewed as different ways of decomposing finite element spaces into sums of subspaces. For the estimation of the constants C_1 and C_2 for different type of decomposition of finite element methods for linear problems, one can find the proofs or references in Xu [29].

Later, the error reduction factor for the above two algorithms shall be estimated. In the following, we shall use $e^n, n = 0, 1, 2, \cdots$, which is defined as:

$$e^n = |\langle F'(u^n) - F'(u), u^n - u\rangle|^{\frac{1}{2}} ,$$

as a measure of the error between u^n and u. Here and later, u stands for the unique solution of (2.1). For convenience, constants α_{min} and α_{max} are defined as $\alpha_{min} = \min_{1\le i\le m}\alpha_i$, $\alpha_{max} = \max_{1\le i\le m}\alpha_i$, and α_i is the relaxation parameters in Algorithm 2.1. Constants C_p and C_s, which are

$$C_p = (\alpha_{min}^{-\frac{1}{2}}L + \alpha_{max}^{\frac{1}{2}}C_2)C_1, \qquad C_s = (L + C_2)C_1, \tag{2.10}$$

will play an important rule in analysing the error reduction factor.

Remark 2.1.

(1) When F is differentiable and if we define

$$w_i^{n+\frac{1}{2}} = \sum_{k=1,k\neq i}^{m} u_k^n + u_i^{n+\frac{1}{2}}\ , \tag{2.11}$$

then (2.3) is equivalent to solving

$$\left\langle F'(w_i^{n+\frac{1}{2}}), v_i\right\rangle = 0, \quad \forall v_i \in V_i\ . \tag{2.12}$$

(2) Let

$$u^{n+1} = \sum_{i=1}^{m} u_i^{n+1},\ n = 0, 1, 2, \cdots \tag{2.13}$$

and $w_i^{n+\frac{1}{2}}$ be defined as in (2.11), then

$$w_i^{n+\frac{1}{2}} = u^n + u_i^{n+\frac{1}{2}} - u_i^n$$

and the value of u^{n+1} corresponding to (2.4) can be obtained by

$$\begin{aligned} u^{n+1} &= \sum_{i=1}^{m} u_i^n + \sum_{i=1}^{m}\alpha_i(u_i^{n+\frac{1}{2}} - u_i^n) \\ &= u^n + \sum_{i=1}^{m}\alpha_i(u_i^{n+\frac{1}{2}} - u_i^n) \\ &= \sum_{i=1}^{m}\alpha_i(u^n + u_i^{n+\frac{1}{2}} - u_i^n) + (1 - \sum_{i=1}^{m}\alpha_i)u^n \\ &= \sum_{i=1}^{m}\alpha_i w_i^{n+\frac{1}{2}} + (1 - \sum_{i=1}^{m}\alpha_i)u^n. \end{aligned} \tag{2.14}$$

In the applications of [24], with a two level domain decomposition method, only the values of u^{n+1} and the coarse mesh problem are needed for the next iteration, and u^{n+1} is updated by the above formula after the computations of each of subdomain problems.

(3) For Algorithm 2.2, if we define

$$w_i^{n+1} = \sum_{k<i} u_k^{n+1} + u_i^{n+1} + \sum_{k>i} u_k^n\ , \tag{2.15}$$

then it satisfies

$$\left\langle F'(w_i^{n+1}), v_i\right\rangle = 0, \quad \forall v_i \in V_i\ . \tag{2.16}$$

and after the solving of w_i^{n+1} from (2.16) for each i, we only need to set $u^{n+1} = w_m^{n+1}$.

(4) Intuitively, one may think that the algorithms need rather large amount of memory. However, in the implementation later for a two level domain decomposition, we only need to store the value of u^{n+1} and one of the $w_i^{n+\frac{1}{2}}$ (the coarse mesh solution) in the memory.

Remark 2.2. Algorithm 2.2 solves the minimization problems sequentially over each subspace. Algorithm 2.1 solves the minimizations in parallel over each of the subspaces. In applications, by suitably decomposing the minimization space, the minimization problem over each subspace can be done by many parallel processors, and so both algorithms are suitable for parallel machines, see [24]. Moreover, with a suitable decomposition, the constant C_1 can be made to be independent of the size of the problem, and so the convergence of the above two algorithms also does not depend on the size of the problem.

3. The convergence of the algorithms

We first give the rate of convergence for Algorithm 2.1.

Theorem 3.1. *If the space decomposition satisfies (2.8),(2.9) and the function satisfies (2.6), then for Algorithm 2.1 we have:*

(a). *If F is quadratic with respect to v, there holds*

$$|e^{n+1}|^2 \leq \frac{C_p^2}{1+C_p^2}|e^n|^2 \ , \forall n \geq 0 \ . \tag{3.1}$$

(b). *If F is third order continuously differentiable, then*

$$|e^{n+1}| \to 0 \ as \ n \to \infty \ , \ and \quad |e^{n+1}|^2 \leq \beta_n |e^n|^2 \ , \quad \forall n \geq 0 \ .$$

For n sufficiently large, we have $0 < \beta_n < 1$. In fact

$$\lim_{n\to\infty} \beta_n = \frac{C}{1+C} < 1 \quad and \quad C = \frac{{C_p}^2}{K^2} \ . \tag{3.2}$$

The convergence of Algorithm 2.2 is similar as Algorithm 2.1.

Theorem 3.2. *Let the space decomposition satisfies (2.8) and the function satisfies (2.7), then for Algorithm 2.2 we have:*

(a). *If F is quadratic with respect to v, there holds*

$$|e^{n+1}|^2 \leq \frac{{C_s}^2}{1+{C_s}^2}|e^n|^2 \ , \forall n \geq 0 \ . \tag{3.3}$$

(b). *If F is third order continuously differentiable, then*

$$|e^{n+1}| \to 0 \ as \ n \to \infty \ , \ and \quad |e^{n+1}|^2 \leq \beta_n |e^n|^2 \ , \quad \forall n \geq 0 \ .$$

For n sufficiently large, we have $0 < \beta_n < 1$. In fact

$$\lim_{n\to\infty} \beta_n = \frac{C}{1+C} < 1 \quad and \quad C = \frac{{C_s}^2}{K^2} \ . \tag{3.4}$$

Acknowledgement. The authors like to thank J. Xu and P. Bjorstad. Many insightful comments from J. Xu during the process of the work helped us to improve some of the results and the presentation of the paper. Some discussions with and valuable comments by P. Bjorstad clarify the relationship of our methods with the literature results.

References

1. O. Axellsson and I. E. Kaporin, *Minimum residual adaptive multilevel procedure for the finite element solution of nonlinear stationary problems*, Preprint, 1995.
2. R. Bank, *Analysis of a multilevel iterative method for nonlinear finite element equations*, Math. Comp. . 39 (1982), 453-465.
3. P. Bjorstad and O. Widlund, *Iterative methods for the solution of elliptic problems on regions portioned into substructuring*, SIAM J. Numer. Anal. . 23 (1986), 1097–1120.
4. J. H. Bramble, R. Ewing, J. E. Pasciak and A. H. Schatz, *A preconditioning technique for the efficient solution of problems with local grid refinement*, Comput. Meth. Appl. Mech. Eng. 67 (1988), 149-159.
5. J. H. Bramble, J. E. Pasciak and A. H. Schatz, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp. 46 (1986), 361–369.
6. ———, *The construction of preconditioners for elliptic problems by substructuring. I.*, Math. Comp. . 47 (1986), 103-134.
7. ———, *The construction of preconditioners for elliptic problems by substructuring. IV*, Math. Comp. 53 (1989), 1–24.
8. A. Brandt, *Multilevel adaptive solutions to boundary value problems*, Math. Comp. . 31 (1977), 333-309.

9. X.-C. Cai and M. Dryja, *Domain decomposition methods for monotone nonlinear elliptic problems*, Proceeding of the seventh international conference on domain decomposition methods in Science and scientific computing (Penn. state Univ.) (D. E. Keyes and Xu, eds.), AMS, Providence, 1994, pp. 21–28.
10. X. C. Cai and O. B. Widlund, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems*, SIAM J. Numer. Anal. **30** (1993), 936-952.
11. J. Cea, *Optimisation – thérie et algorithmes*, Dunod, 1971.
12. C. N. Dawson and M. F. Wheeler, *Two-grid methods for mixed finite element approximations of nonlinear parabolic equations*, Seventh international conference on domain decomposition methods in Science and scientific computing (Penn. state Univ.) (D. E. Keyes and Xu, eds.), AMS, Providence, 1994, pp. 191–203.
13. M. Dryja, B. Smith and O. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal. . **31** (1994), 1662-1694.
14. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comp. **15** (1994), 604-620.
15. I. Ekeland and R. Temam, *Convex analysis and variational problems*, North–Holland, Amsterdam, 1976.
16. M. Espedal and R. Ewing, *Characteristic Petrov-Galerkin subdomain methods for two phase immiscible flow*, Comput. Methods in Appl. Mech. and Engig. . **64** (1987), 113-135.
17. W. Hackbush, *Iterative methods for large sparse linear systems*, Springer, Heidelberg, 1993.
18. K. Kunisch and X.-C. Tai, *Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces*, To appear in SIAM J. Numer. Anal. (Preprint, Technische Universität Graz, July 1993).
19. ———, *Domain decomposition methods for elliptic parameter estimation problems*, preprint (1995).
20. P. L. Lions, *On the Schwarz alternating method II,*, Domain decomposition methods for partial differential equations,II (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM, Philadelphia, 1989, pp. 47-70.
21. L. D. Marini and A. Quarteroni, *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math. **55** (1989), 575–598.
22. S. McCormick, *Multilevel adaptive methods for partial differential equations*, SIAM (Philadelphia, PA), 1989.
23. R. Rannacher, *On the convergence of the Newton-Raphson method for strongly nonlinear finite element equations*, Nonlinear computational mechanics (P. Wriggers and W. Wagner, eds.), Springer-Verlag, 1991.
24. X.-C. Tai and M. Espedal, *Rate of convergence of a two level domain decomposition method*, Submitted to the proceeding of the 8th domain decomposition conference (1995).
25. X.–C. Tai, *Parallel function decomposition and space decomposition methods with applications to optimisation, splitting and domain decomposition*, Preprint No. 231–1992, Institut für Mathematik, Technische Universität Graz (1992).
26. ———, *Parallel function and space decomposition methods.*, Finite element methods, fifty years of the courant element (P. Neittaanmäki, eds.), Lecture notes in pure and applied mathematics, vol. 164, Marcel Dekker inc, 1994, pp. 421-432.
27. ———, *Domain decomposition for linear and nonlinear elliptic problems via function or space decomposition*, Domain decomposition methods in scientific and engineering computing (Proc. of the 7th international conference on domain decomposition, Penn. State University, 1993) (D. Keyes and J. Xu, eds.), American Mathematical Society, 1995, pp. 355-360.
28. B. F. Smith, P. E. Bjorstad and W. D. Gropp, *Domain decomposition: Parallel multilevel algorithms for elliptic partial differential equations*, Combridge Univ. Press, Cambridge, 1995 (toappear).
29. J. C. Xu, *Iteration methods by space decomposition and subspace correction*, SIAM Rev. **34** (1992), 581-613.
30. ———, *A novel two-grid method for semilinear elliptic equations*, SIAM J. Sci. Comp. . **15** (1994), 231-237.
31. ———, *Two-grid discretization techniques for linear and nonlinear PDE*, Tech. report AM105, Depart. of Math., Penn. State University, University park, July, 1992.
32. X. Zhang, *Multilevel Schwarz methods*, Numer. Math. . **63** (1992), 521-539,.

Topic:	*Session Chair:*	*Room B*
Krylov Methods	*Anne Greenbaum*	

4:45 - 5:15	E. Bobrovnikova	Iterative Methods for Weighted Least-Squares
5:15 - 5:45	A. Lumsdaine	Krylov Subspace Acceleration of Waveform Relaxation
5:45 - 6:15	C. Wagner	Tangential Frequency Filtering Decompositions
6:15 - 6:45	J. Zhang	Multigrid Solution of Convection-Diffusion Equation with High-Reynolds Number

Iterative Methods for Weighted Least-squares*

Elena Y. Bobrovnikova† Stephen A. Vavasis‡

January 15, 1996

Abstract. A weighted least-squares problem with a very ill-conditioned weight matrix arises in many applications. Because of round-off errors, the standard conjugate gradient method for solving this system does not give the correct answer even after n iterations. In this paper we propose an iterative algorithm based on a new type of reorthogonalization that converges to the solution.

Consider the linear least-squares system

$$\min \|D^{1/2}(\mathbf{b} - A\mathbf{x})\|_2 \tag{1}$$

where $D \in \mathbf{R}^{m\times m}, A \in \mathbf{R}^{m\times n}, \mathbf{b} \in \mathbf{R}^m$.

We make the following assumptions: D is a symmetric, positive definite matrix; $\operatorname{rank} A = n$. These assumptions imply that (1) is a nonsingular linear system with a unique solution.

This problem arises in many applications, including interior point methods, electrical networks, finite element methods and structures. Matrix D is, respectively, objective function Hessian, electrical resistances (their reciprocals), thermal conductivity and element flexibility.

Our paper is focused on the case when matrix D is severely ill-conditioned. This happens in certain classes of finite element problems [6], electrical networks and always occurs in optimization involving a barrier function. In interior-point methods, matrix D becomes very ill-conditioned as iterates approach the boundary of the feasible region [8]. In linear programming, since the solution is always on the boundary of the region, ill-conditioning always occurs during the algorithm. In many settings D is diagonal

*This work has been supported by an NSF Presidental Young Investigator grant, with matching funds received from AT&T and Xerox Corp.

†Center for Applied Mathematics, Cornell University, Ithaca, New York 14853.

‡Center for Applied Mathematics and Department of Computer Science, Cornell University, Ithaca, New York 14853.

or can be made diagonal by some transformation. Hence we also assume that D is a diagonal matrix.

Normal equations for (1) have the form

$$A^tDA\mathbf{x} = A^tD\mathbf{b}. \tag{2}$$

Because of ill-conditioning of the system, standard methods such as QR factorization, Cholesky factorization, symmetric indefinite factorization, range-space method and null-space method are unstable. Here *stability* of an algorithm, as defined by Vavasis [7], means that the computed solution $\hat{\mathbf{x}}$ satisfies the error bound

$$\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \epsilon \cdot f(A) \cdot \|\mathbf{b}\|, \tag{3}$$

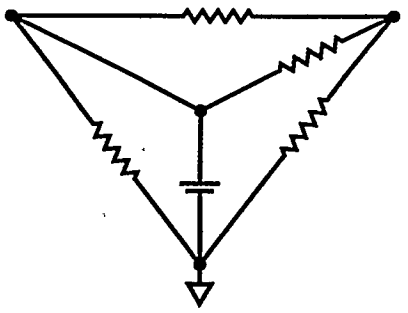
where ϵ is machine precision and $f(A)$ is some function of A not depending on D. Recently Vavasis [7] and Hough and Vavasis [3] proposed stable direct methods for (1). We would like to have iterative methods for this problem because they are much more efficient than direct for large sparse problems, which is the common setting in applications.

It is well known that the standard unpreconditioned conjugate-gradient (CG) method converges in no more than n iterations in exact arithmetic. In the presence of finite precision arithmetic, however, it may converge slowly or not at all. Consider a four-node electrical network composed of resistors and batteries (see Figure 1). Ohm's and Kirchhoff's laws can be applied to formulate a 3×3 linear system of the form (2). After 3 iterations of standard CG, only two correct significant digits are obtained (measured in a forward error sense $\|\mathbf{x} - \hat{\mathbf{x}}\|$), and further iterations do not improve the solution. In contrast, the circuit is sufficiently simple that it is possible by visual inspection to determine the correct solution to 15 digits of accuracy.

In this paper we propose an iterative method based on the conjugate gradient method with a new type of reorthogonalization and give some numerical evidence of its stability. For the background on reorthogonalization see [2], [4], [5].

The main ideas of the new method are as follows. First, we introduce new sequences of vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots$ and $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$ lying in $\mathbf{R}^m$. They are related to search directions and residuals of CG method according to equations below. From these vectors we can determine whether there is a catastrophic cancellation in CG procedure. We can also restore CG convergence by correcting the entries of these vectors that are affected by cancellation. Correcting requires solving linear systems. We show below, however, that we can efficiently solve these systems using information already computed by CG.

Let the search directions of CG applied to (2) be denoted $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots$ and the residuals $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots$. Let C denote A^tDA. In exact arithmetic the CG procedure implies orthogonality of residuals $\mathbf{r}^{(k)t}\mathbf{r}^{(l)} = 0$ and conjugacy of search directions $\mathbf{p}^{(k)t}C\mathbf{p}^{(l)} = 0$.



(a)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 10^{15} & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 10^{15} & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

(b)

Figure 1: *The circuit depicted in (a) yields an weighted least-squares system of the form (2) with D, A, b as in (b). The correct values of voltages of the three nodes are* $(1.00, 1.00, 0.67)$ *accurate to two digits. The standard CG algorithm, however, yields* $(1.00, 1.00, 0.00)$.

Let us introduce new variables $\mathbf{v}^{(k)}$ and $\mathbf{w}^{(k)}$,

$$\mathbf{v}^{(k)} = DA\mathbf{p}^{(k)}, k = 1, 2, ... \quad (4)$$

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \alpha_k A\mathbf{p}^{(k)}, k = 1, 2, ..., \quad (5)$$

$$\mathbf{w}^{(0)} = \mathbf{b}.$$

It is easy to see that $\mathbf{r}^{(k)} = A^t D\mathbf{w}^{(k)}$.

To simplify discussion, we start with the case when the entries of D have two distinct scales. Suppose that k entries of D are large, of the same order and correspond to submatrix of A of row rank k. The remaining entries of D are much smaller and approximately equal. Finally, assume $\mathbf{b}$ is 'generic'. Our algorithm is based on the following observations. For the first k iterations conjugate directions have components corresponding to large entries of D much larger than the rest (i.e. $\mathbf{p}$'s 'almost lie' in the subspace corresponding to large components of D). After k iterations of the CG method, the corresponding k entries of $\mathbf{w}$ are close to zero. Multiplied by large entries of D they cause cancellation in the computation of the residual. In other words, after k steps standard CG does not accurately determine the component of $\mathbf{r}^{(k)}$ parallel to these directions. Hence the next search direction is not computed correctly. Direction $\mathbf{p}$ after k iterations is almost orthogonal to the subspace of large directions. The small

components of $\mathbf{p}$, multiplied by large components of D, cause miscalculation of $\mathbf{v}^{(k)}$, hence of α_k. To solve this problem we correct components of $\mathbf{v}$ and $\mathbf{w}$ corresponding to large entries of D using orthogonality and conjugacy relations.

Orthogonality relations on step j against the first k residual vectors have the form $R_k^t \mathbf{r}^{(j)} = \mathbf{0}$, where $R_k = \left(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, ..., \mathbf{r}^{(k)}\right)$, or $R_k^t A^t D \mathbf{w}^{(j)} = \mathbf{0}$.

Let us denote $\bar{\mathbf{w}}^{(j)} \in \mathbf{R}^k$ the entries of $\mathbf{w}$ corresponding to the large entries of D, $\hat{\mathbf{w}}^{(j)} \in \mathbf{R}^{m-k}$ the remaining entries of $\mathbf{w}$, $\bar{A} \in \mathbf{R}^{k \times n}$, $\hat{A} \in \mathbf{R}^{(m-k) \times n}$ and $\bar{D} \in \mathbf{R}^{k \times k}$, $\hat{D} \in \mathbf{R}^{(m-k) \times (m-k)}$ the corresponding submatrices of A and D. With this notation the orthogonality relation becomes

$$R_k^t \bar{A}^t \bar{D} \bar{\mathbf{w}}^{(j)} = -R_k^t \hat{A}^t \hat{D} \hat{\mathbf{w}}^{(j)} := \mathbf{d},$$

where we assume that $\mathbf{d}$ is accurately computed. Assume $\bar{\mathbf{w}}^{(j)} = \bar{A} R_k \bar{\mathbf{y}}$ for some $\bar{\mathbf{y}}$. Then the last equation takes the form

$$R_k^t \bar{A}^t \bar{D} \bar{A} R_k \bar{\mathbf{y}} = \mathbf{d}. \tag{6}$$

In exact arithmetic (6) holds at each step of CG. Our correction step involves solving (6) for $\bar{\mathbf{y}}$ to yield $\bar{\mathbf{w}}^{(j)}$, thus accurately computing the entries of $\bar{\mathbf{w}}^{(j)}$ that were contaminated by cancellation.

Similarly, the conjugacy relationship on step j has the form

$$P_k^t A^t \mathbf{v}^{(j)} = \mathbf{0}, P_k = \left(\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, ..., \mathbf{p}^{(k)}\right)$$

or

$$Q_k^t A^t \mathbf{v}^{(j)} = \mathbf{0}, Q_k = \left(\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, ..., \mathbf{q}^{(k)}\right),$$

where $\mathbf{q}^{(l)} = \mathbf{r}^{(l)} / \|\mathbf{r}^{(l)}\|$. The last relationship holds because $\mathbf{q}$'s and $\mathbf{p}$'s span the same Krylov space. Denoting $\bar{\mathbf{v}}^{(j)}$ the entries of $\mathbf{v}^{(j)}$ corresponding to large components of D and $\hat{\mathbf{v}}^{(j)}$ the remaining entries of $\mathbf{v}^{(j)}$ and assuming $\bar{\mathbf{v}}^{(j)} = \bar{D} \bar{A} Q_k \bar{\mathbf{z}}$ for some $\bar{\mathbf{z}}$ we get

$$Q_k^t \bar{A}^t \bar{D} \bar{A} Q_k \bar{\mathbf{z}} = \mathbf{f}. \tag{7}$$

Recall that the CG method essentially computes a tridiagonal factorization of $A^t D A$: $A^t D A \approx Q_k T_k Q_k^t$, and $T_k = L_k S_k L_k^t$, where S_k is diagonal and L_k is unit lower bidiagonal. Since matrices of systems (6) and (7) are close to tridiagonal matrices $R_k^t A^t D A R_k$ and $Q_k^t A^t D A Q_k$, respectively, whose factorization we already computed by step j, we can solve these systems by iterative refinement. The reason these systems are close is precisely because the large entries of D are in $\bar{D}$.

Recall that iterative refinement can be used to solve a linear system $K\mathbf{x} = \mathbf{c}$ provided there exists a matrix $\bar{K}$ close to K such that linear systems involving $\bar{K}$ are easy to solve (see [1]).

The first correction occurs on step when norm of the residual drops by several orders of magnitude (for a generic $\mathbf{b}$). This happens on the step equal to rank of rows of A corresponding to large entries of D.

Suppose now that matrix D has entries of three different scales and $\mathbf{b}$ is generic. Then norm of the residual drops dramatically twice during the computations. After the first drop of the residual on step k we correct entries of $\mathbf{w}$ and $\mathbf{v}$ corresponding to 'large' entries of D by solving equations (6) and (7). After the second drop of the residual on step l we correct entries of $\mathbf{w}$ and $\mathbf{v}$ corresponding to 'large' and 'medium' entries of D.

Let R_k and R_l denote matrices of the residuals $\left(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, ..., \mathbf{r}^{(k)}\right)$ and $\left(\mathbf{r}^{(k+1)}, \mathbf{r}^{(k+2)}, ..., \mathbf{r}^{(l)}\right)$, respectively. Let $\bar{\mathbf{w}}^{(j)} \in \mathbf{R}^k$ denote the entries of $\mathbf{w}$ corresponding to the large entries of D on step j, $\tilde{\mathbf{w}}^{(j)} \in \mathbf{R}^{(l-k)}$ the entries corresponding to the medium entries of D on step j and $\hat{\mathbf{w}}^{(j)} \in \mathbf{R}^{m-l}$ the remaining entries of $\mathbf{w}$, $\bar{A} \in \mathbf{R}^{k \times n}$, $\tilde{A} \in \mathbf{R}^{(l-k)\times n}$, $\hat{A} \in \mathbf{R}^{(m-l)\times n}$ and $\bar{D} \in \mathbf{R}^{k\times k}$, $\tilde{D} \in \mathbf{R}^{(l-k)\times(l-k)}$, $\hat{D} \in \mathbf{R}^{(m-k)\times(m-k)}$ the corresponding submatrices of A and D. Then the orthogonality relationship becomes

$$\begin{cases} R_k^t \left(\bar{A}^t \bar{D} \bar{\mathbf{w}}^{(j)} + \tilde{A}^t \tilde{D} \tilde{\mathbf{w}}^{(j)}\right) = -R_k^t \hat{A}^t \hat{D} \hat{\mathbf{w}}^{(j)} := \mathbf{d}_k \\ R_l^t \left(\bar{A}^t \bar{D} \bar{\mathbf{w}}^{(j)} + \tilde{A}^t \tilde{D} \tilde{\mathbf{w}}^{(j)}\right) = -R_l^t \hat{A}^t \hat{D} \hat{\mathbf{w}}^{(j)} := \mathbf{d}_l \end{cases} \quad (8)$$

Assume $\bar{\mathbf{w}}^{(j)} = \bar{A} R_k \bar{\mathbf{y}}$ and $\tilde{\mathbf{w}}^{(j)} = \tilde{A} R_l \tilde{\mathbf{y}}$ for some $\bar{\mathbf{y}}$ and $\tilde{\mathbf{y}}$. Then (8) becomes a system with block matrix

$$\begin{pmatrix} R_k^t \bar{A}^t \bar{D} \bar{A} R_k & R_k^t \tilde{A}^t \tilde{D} \tilde{A} R_l \\ R_l^t \bar{A}^t \bar{D} \bar{A} R_k & R_l^t \tilde{A}^t \tilde{D} \tilde{A} R_l \end{pmatrix}.$$

Similarly, the conjugacy relationship on step j leads to a system of equations

$$\begin{cases} Q_k^t \left(\bar{A}^t \bar{D} \bar{\mathbf{v}}^{(j)} + \tilde{A}^t \tilde{D} \tilde{\mathbf{v}}^{(j)}\right) = -Q_k^t \hat{A}^t \hat{D} \hat{\mathbf{v}}^{(j)} := \mathbf{f}_k \\ Q_l^t \left(\bar{A}^t \bar{D} \bar{\mathbf{v}}^{(j)} + \tilde{A}^t \tilde{D} \tilde{\mathbf{v}}^{(j)}\right) = -Q_l^t \hat{A}^t \hat{D} \hat{\mathbf{v}}^{(j)} := \mathbf{f}_l \end{cases} \quad (9)$$

which, assuming $\bar{\mathbf{v}}^{(j)} = \bar{A} Q_k \bar{\mathbf{z}}$ and $\tilde{\mathbf{v}}^{(j)} = \tilde{A} Q_l \tilde{\mathbf{z}}$ for some $\bar{\mathbf{z}}$ and $\tilde{\mathbf{z}}$, has block structure

$$\begin{pmatrix} Q_k^t \bar{A}^t \bar{D} \bar{A} Q_k & Q_k^t \tilde{A}^t \tilde{D} \tilde{A} Q_l \\ Q_l^t \bar{A}^t \bar{D} \bar{A} Q_k & Q_l^t \tilde{A}^t \tilde{D} \tilde{A} Q_l \end{pmatrix}.$$

The norms of low-triangular blocks $R_l^t \bar{A}^t \bar{D} \bar{A} R_k$ and $Q_l^t \bar{A}^t \bar{D} \bar{A} Q_k$ are much smaller than the norms of diagonal blocks. Hence instead of solving (8) and (9) we solve systems with upper block-triangular matrices. Similarly to the case of entries of D of two scales, matrices $R_k^t \bar{A}^t \bar{D} \bar{A} R_k$, $R_l^t \tilde{A}^t \tilde{D} \tilde{A} R_l$ and $Q_k^t \bar{A}^t \bar{D} \bar{A} Q_k$ $Q_l^t \tilde{A}^t \tilde{D} \tilde{A} Q_l$ are close to tridiagonal matrices $R_k^t A^t D A R_k$, $R_l^t A^t D A R_l$ and $Q_k^t A^t D A Q_k$, $Q_l^t A^t D A Q_l$ whose factorization we already computed by the CG method. Hence we can approximately solve systems (8) and (9) using iterative refinement.

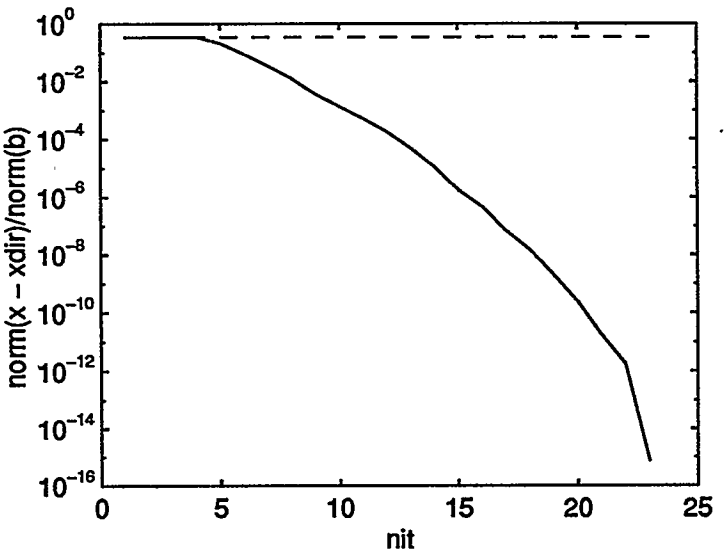


Figure 2: *Modified (solid line) and standard (dashed line) CG method for a boundary-value problem with matrix D with three different scales. Relative accuracy of the modified CG method is* $7.2 \cdot 10^{-16}$

In order to generalize the proposed method to the case when $\mathbf{b}$ is not generic we consider instead of system (2) the system with right-hand side $\mathbf{b}_r = \mathbf{b} + A\mathbf{x}_r$, where $\mathbf{x}_r$ is a random perturbation of the size $\|\mathbf{b}\|/\|A\|$. After finding the solution to this modified system we subtract the random component $\mathbf{x}_r$ from it.

Our ultimate goal is to establish (3) theoretically, but currently our results are of an experimental nature. We tested this method on a boundary-value problem with matrix A of dimension 136×23 and compared our results with solution by the direct method given by [3]. The matrix A in question has many rank-deficient submatrices. For all considered matrices D and right-hand sides $\mathbf{b}$ we got relative accuracy of order 10^{-14} to 10^{-16} after n iterations (see Figure 2). In contrast, plain CG gave very poor answers. It would be interesting to try to improve the rate of convergence by using an appropriate preconditioner.

References

[1] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Second edition, 1989.

[2] G. H. Golub, R. Underwood, and J. H. Wilkinson. The Lanczos algorithm for the symmetric $Ax = \lambda Bx$ problem. Technical Report STAN-CS-72-270, Department of Computer Science, Stanford University, Stanford, California, 1972.

[3] P. D. Hough and S. A. Vavasis. Complete orthogonal decomposition for weighted least squares. Technical Report CTC94TR203, Cornell Theory Center, 1994.

[4] C. C. Paige. Practical use of symmetric Lanczos process with reorthogonalization. *BIT*, 10:183–195, 1970.

[5] B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comp*, 33:217–238, 1979.

[6] S. A. Vavasis. Stable finite elements for problems with wild coefficients. Technical Report TR-93-1364, Department of Computer Science. Cornell University, Ithaca, N.Y., 1993.

[7] S. A. Vavasis. Stable numerical algorithms for equilibrium systems. *SIAM J. Matrix Anal. Appl.*, 15:1108–1131, 1994.

[8] M. H. Wright. Interior methods for constrained optimization. In *Acta Numerica 1992*. Cambridge University Press, Cambridge, 1992.

KRYLOV SUBSPACE ACCELERATION OF WAVEFORM RELAXATION*

ANDREW LUMSDAINE† AND DEYUN WU‡

1. Introduction. Standard solution methods for numerically solving time-dependent problems typically begin by discretizing the problem on a uniform time grid and then sequentially solving for successive time points. The initial time discretization imposes a serialization to the solution process and limits parallel speedup to the speedup available from parallelizing the problem at any given time point. This bottleneck can be circumvented by the use of *waveform methods* in which multiple time-points of the different components of the solution are computed independently.

With the waveform approach, a problem is first spatially decomposed and distributed among the processors of a parallel machine. Each processor then solves its own time-dependent subsystem over the entire interval of interest using previous iterates from other processors as inputs. Synchronization and communication between processors take place infrequently, and communication consists of large packets of information — discretized functions of time (i.e., waveforms).

Unfortunately, the convergence rate of standard waveform relaxation can be prohibitively slow for many problems of interest. Previous approaches for accelerating the convergence of waveform relaxation include the shifted Picard iteration [14], multigrid [1, 15], SOR [9], convolution SOR [11], $\mathbb{L}^2$ Krylov subspace methods [8], and adaptive window size selection [7].

Many of these approaches are similar to acceleration methods for iteratively solving linear systems of equations. However, in most cases, the generalizations of those approaches to waveform relaxation do not accelerate convergence to the same degree as their linear algebra counterparts [9]. An analysis of why acceleration of waveform relaxation can, in general, be expected to be small is given in [10]. An exception to the pessimistic result of [10] is the convolution SOR method developed in [12].

In this paper, we describe and analyze Krylov-subspace methods for accelerating the convergence of waveform relaxation for solving time dependent problems. We first review the $\mathbb{L}^2$ Krylov subspace techniques presented in [8] and then present the Convolution CG and the Convolution GMRES algorithms for linear operators on a Hilbert space. For a model case, we compare the CCG algorithm for linear differential equations with the CG algorithm for linear algebraic equations, and we prove that they have the same convergence rate bound.

2. Waveform Relaxation. A mathematical description of waveform methods can be developed easily through the use of a model initial value problem (IVP):

$$\begin{aligned} \tfrac{d}{dt}\boldsymbol{x}(t) + A\boldsymbol{x}(t) &= \boldsymbol{f}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{f}(t) \in \mathbb{R}^n$, and $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the unknown vector to be computed over an interval of interest, $t \in [0, T]$. The traditional approach for numerically solving the IVP begins by discretizing (2.1) in time with an implicit integration rule (since large dynamical systems are typically stiff) and then solving the resulting matrix problem at each time step [3, 4]. This *pointwise* approach can be disadvantageous for a parallel implementation, especially for distributed memory parallel computers having a high communication latency, since the processors will have to synchronize repeatedly for each timestep.

A more suitable approach to solving the IVP with a parallel computer is to decompose the problem at the differential equation level. That is, the large system is decomposed into smaller subsystems, each of which is assigned to a single processor. The IVP is solved iteratively by solving the smaller IVPs for each subsystem, using fixed values from previous iterations for the variables from other subsystems. This dynamic iteration process is variously known as waveform relaxation (WR), dynamic iteration, or as the Picard-Lindelöf iteration [9, 16].

In (2.1), let $A = M - N$ be a splitting of A. The waveform relaxation algorithm based on this splitting is expressed in matrix form as

ALGORITHM 2.1. [Waveform Relaxation for Linear Systems]

1. *Initialize:* Pick $\boldsymbol{x}^0$
2. *Iterate:* For waveform iteration $k = 0, 1, \ldots$

* This work was supported in part by National Science Foundation grant CCR92-09815.

† Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, Andrew.Lumsdaine.1@nd.edu

‡ Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, deyun.wu.2@nd.edu

$$\text{Solve} \quad \begin{aligned} \tfrac{d}{dt}\boldsymbol{x}^{k+1}(t) + \boldsymbol{M}\boldsymbol{x}^{k+1}(t) &= \boldsymbol{N}\boldsymbol{x}^{k}(t) + \boldsymbol{f}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0 \end{aligned}$$

for $\boldsymbol{x}^{k+1}(t)$ on $[0,T]$.

We can solve for $\boldsymbol{x}^{k+1}(t)$ explicitly [6], that is,

$$\boldsymbol{x}^{k+1}(t) = e^{-\boldsymbol{M}t}\boldsymbol{x}(0) + \int_0^t e^{-\boldsymbol{M}(t-s)}\left(\boldsymbol{N}(s)\boldsymbol{x}^k(s) + \boldsymbol{f}(s)\right) ds. \tag{2.2}$$

Instead of using this formulation, it is useful to abstract (2.2) and consider $\boldsymbol{x}$ as an element of a function space (of n-dimensional functions) and the integral as an operator on n-dimensional functions. Using operator notation, we can write (2.2) as

$$\boldsymbol{x}^{k+1} = \mathcal{K}\boldsymbol{x}^k + \boldsymbol{\psi}. \tag{2.3}$$

Here the variables are defined on the space of n-dimensional square integrable functions, which we will denote as $\mathbb{L}^2([0,T],\mathbb{R}^n)$. The operator $\mathcal{K}$ (mapping from $\mathbb{L}^2([0,T],\mathbb{R}^n)$ to $\mathbb{L}^2([0,T],\mathbb{R}^n)$) is defined by

$$(\mathcal{K}\boldsymbol{x})(t) = \int_0^t e^{-\boldsymbol{M}(t-s)}\boldsymbol{N}(s)\boldsymbol{x}(s)ds,$$

and $\boldsymbol{\psi} \in \mathbb{L}^2([0,T],\mathbb{R}^n)$ is given by

$$\boldsymbol{\psi}(t) = e^{-\boldsymbol{M}t}\boldsymbol{x}(0) + \int_0^t e^{-\boldsymbol{M}(t-s)}\boldsymbol{f}(s)ds.$$

Roughly, application of the operator $\mathcal{K}$ means: "take one step of waveform relaxation."

Now, we also know (based on the splitting) that the solution $\boldsymbol{x}$ to (2.1) will satisfy

$$\begin{aligned} \tfrac{d}{dt}\boldsymbol{x}(t) + \boldsymbol{M}\boldsymbol{x}(t) &= \boldsymbol{N}\boldsymbol{x}(t) + \boldsymbol{f}(t) \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0. \end{aligned}$$

Or, using operator notation, we can see that $\boldsymbol{x}$ will satisfy

$$(\boldsymbol{I} - \mathcal{K})\boldsymbol{x} = \boldsymbol{\psi} \tag{2.4}$$

where the operator $\boldsymbol{I}$ is the identity operator.

3. Hilbert Space Methods. It has been known since the early development of Krylov-subspace iterative methods for linear algebra that the methods are easily extended to Hilbert space [5]. A natural Hilbert space for this problem is $\mathbb{L}^2([0,T],\mathbb{R}^n)$.

Since $\mathcal{K}$ is not self-adjoint in $\mathbb{L}^2([0,T],\mathbb{R}^n)$, we can apply waveform GMRES (WGMRES), which is an extension of GMRES to the space $\mathbb{L}^2([0,T],\mathbb{R}^n)$. The WGMRES algorithm is given as follows:

ALGORITHM 3.1. [Waveform GMRES]

1. *Start:* Set $r^0 = \psi - (I - \mathcal{K})x^0$, $v^1 = r^0/\|r^0\|$, $\beta = \|r^0\|$.
2. *Iterate:* For $k = 1, 2, \ldots$, until satisfied do:
   - $h_{j,k} = \langle (I - \mathcal{K})v^k, v^j \rangle,\ j = 1, 2, \ldots, k$
   - $\hat{v}^{k+1} = (I - \mathcal{K})v^k - \sum_{j=1}^{k} h_{j,k}v^j$
   - $h_{k+1,k} = \|\hat{v}^{k+1}\|$
   - $v^{k+1} = \hat{v}^{k+1}/h_{k+1,k}$.
3. *Form approximate solution:*
   - $x^k = x^0 + V^k y^k$, where y^k minimizes $\|\beta e_1 - \bar{H}^k y^k\|$.

4. Convolution Methods. Corresponding to the IVP (2.1), there is a linear algebraic equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{4.1}$$

It is well-known that SOR for linear algebraic equation (4.1) converges very well. But waveform SOR for linear differential equations (2.1) does not converge as well as SOR does except for some special cases [10]. Recently, [12]

gave an optimal convolution SOR (CSOR) to accelerate the convergence of WSOR such that the CSOR converges as well as SOR does.

A similar approach can be used to develop a convolution conjugate gradient method (CCG). Before we give CCG algorithm, let's give some definitions about convolution and deconvolution.

Definition. Assume $f \in \mathbb{L}^2(-\infty, +\infty)$ is a function, $u,\ v \in \mathbb{L}^2((-\infty, +\infty), \mathbb{R}^n)$ are $\mathbb{L}^2$ vectors. Define

$$\begin{aligned}(f * u)(t) &= ((f * u_i)(t))^T \in \mathbb{L}^2((-\infty, +\infty), \mathbb{R}^n),\\ (u * v)(t) &= \textstyle\sum_{i=1}^n (u_i * v_i)(t) \in \mathbb{L}^2(-\infty, +\infty),\end{aligned}$$

where $u = (u_1, u_2, \ldots, u_n)^T$, $v = (v_1, v_2, \ldots, v_n)^T$.

Definition. A $\mathbb{L}^2$ function h is called a deconvolution of f and g, if $f = h * g$, and $g \neq 0$. Denoted it by

$$h = \frac{f}{g}.$$

Notice that if g is compactly supported, then a deconvolution of f and g is unique. In fact, assume

$$f = h_1 * g = h_2 * g.$$

By taking Fourier transform, we get

$$\widehat{f} = \widehat{h_1}\widehat{g} = \widehat{h_2}\widehat{g},$$

which implies that $(\widehat{h_1} - \widehat{h_2})\widehat{g} \equiv 0$. Now, since $supp(g)$ is compact, $\widehat{g}$ is holomorphic and non-zero, we conclude that $\widehat{h_1} = \widehat{h_2}$, *a.e.* Therefore,

$$h_1 = h_2 = \widehat{\widehat{f}/\widehat{g}}$$

in $\mathbb{L}^2$ sense.

Remark. If $f \in \mathbb{L}^2(-\infty, +\infty)$ is not the zero function, then $f * f \neq 0$.

The CCG algorithm is given as follows when an operator $\mathcal{A}$ is convolution self-adjoint from $\mathbb{L}^2([0, T], \mathbb{R}^n)$ to $\mathbb{L}^2([0, T], \mathbb{R}^n)$. Here "convolution self-adjoint" means the following: for any $u,\ v \in \mathbb{L}^2([0, T], \mathbb{R}^n)$,

$$(\mathcal{A}u * v) = (u * \mathcal{A}v).$$

ALGORITHM 4.1. [Convolution CG Algorithm]

1. *Start:* Compute $r_0 = f - \mathcal{A}x_0$, $p_0 = r_0$
2. *Iterate:* For $j = 0, 1, \ldots$ until converged,
   - $\alpha_j = \frac{(r_j * r_j)}{(\mathcal{A}p_j * p_j)}$
   - $x_{j+1} = x_j + \alpha_j * p_j$
   - $r_{j+1} = r_j - \alpha_j * \mathcal{A}p_j$
   - $\beta_j = \frac{(r_{j+1} * r_{j+1})}{(r_j * r_j)}$
   - $p_{j+1} = r_{j+1} + \beta_j * p_j$.

Remark.

1. In the above algorithm, "division" means deconvolution.
2. If $\mathcal{A}$ is a linear elliptic operator, then it is convolution self-adjoint so CCG can be applied.
3. If matrix A is symmetric and $M = dI$, $d > 0$, then the operator $\mathcal{K}$ is convolution self-adjoint.

5. Convergence of CCG Algorithm. It is well known that the convergence rate of CG applied to (4.1) is controlled by

$$\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1},$$

where $\kappa(A) = \lambda_{\max}/\lambda_{\min}$ is the condition number of A (cf. [13]).

By using Fourier transform techniques and results from [2], we can prove the following main theorem of this paper.

THEOREM 5.1. *If $M = dI$, $d > 0$, A is positive definite Hermitian, then in $\mathbb{L}^2$ norm, the CCG algorithm for operator $(I - K)$ in (2.4) has a convergence rate controlled by*

$$\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}.$$

REFERENCES

[1] C. LUBICH AND A. OSTERMAN, *Multigrid dynamic iteration for parabolic problems*, BIT, 27 (1987), pp. 216–234.

[2] V. FABER AND T. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.

[3] C. W. GEAR, *Numerical Initial Value Problems in Ordinary Differential Equations*, Automatic Computation, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[4] E. HAIRER, S. P. NORSETT, AND G. WANNER, *Solving Ordinary Differential Equations*, vol. 1 and 2, Springer-Verlag, New York, 1987.

[5] R. M. HAYES, *Iterative methods of solving linear problems on Hilbert space*, in Contributions to the Solution of Systems of Linear Equations and the Determination of Eigenvalues, O. Taussky, ed., vol. 39 of Nat. Bur. of Standards Applied Math. Series, U.S. Govt. Printing Office, Washington, D.C., 1954, pp. 71–103.

[6] T. KAILATH, *Linear Systems*, Prentice-Hall, Englewood Cliffs, 1980.

[7] B. LEIMKUHLER, *Estimating waveform relaxation convergence*, SIAM J. Sci. Comput., 14 (1993), pp. 872–889.

[8] A. LUMSDAINE, *Theoretical and Practical Aspects of Parallel Numerical Algorithms for Initial Value Problems, with Applications*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[9] U. MIEKKALA AND O. NEVANLINNA, *Convergence of dynamic iteration methods for initial value problems*, SIAM J. Sci. Stat. Comp., 8 (1987), pp. 459–467.

[10] O. NEVANLINNA, *Linear acceleration of Picard-Lindelöf iteration*, Numer. Math., 57 (1990), pp. 147–156.

[11] M. REICHELT, *Accelerated Waveform Relaxation Techniques for the Parallel Transient Simulation of Semiconductor Devices*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[12] M. REICHELT, J. WHITE, AND J. ALLEN, *Optimal convolution sor acceleration of waveform relaxation with application to parallel simulation of semiconductor devices*, SIAM J. Sci. Comput., 16 (1995), pp. 1137–1158.

[13] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS, 1996.

[14] R. D. SKEEL, *Waveform iteration and the shifted Picard splitting*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 756–776.

[15] S. VANDEWALLE AND R. PIESSENS, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1330–1346.

[16] J. K. WHITE AND A. SANGIOVANNI-VINCENTELLI, *Relaxation Techniques for the Simulation of VLSI Circuits*, Engineering and Computer Science Series, Kluwer Academic Publishers, Norwell, Massachusetts, 1986.

Frequency filtering decompositions for unsymmetric matrices
and matrices with strongly varying coefficients

Christian Wagner

In 1992, Wittum introduced the frequency filtering decompositions (FFD), which yield a fast method for the iterative solution of large systems of linear equations. Based on this method, the tangential frequency filtering decompositions (TFFD) have been developed. The TFFD allow the robust and efficient treatment of matrices with strongly varying coefficients. The existence and the convergence of the TFFD can be shown for symmetric and positive definite matrices. For a large class of matrices, it is possible to prove that the convergence rate of the TFFD and of the FFD is independent of the number of unknowns.
For both methods, schemes for the construction of frequency filtering decompositions for unsymmetric matrices have been developed. Since, in contrast to Wittums's FFD, the TFFD needs only one test vector, an adaptive test vector can be used. The TFFD with respect to the adaptive test vector can be combined with other iterative methods, e.g. multi-grid methods, in order to improve the robustness of these methods. The frequency filtering decompositions have been successfully applied to the problem of the decontamination of a heterogeneous porous medium by flushing.

Multigrid Solution of the Convection-Diffusion Equation with High-Reynolds Number*

JUN ZHANG†
Department of Mathematics, The George Washington University, Washington, DC 20052, USA

March 10, 1996

Abstract

A fourth-order compact finite difference scheme is employed with the multigrid technique to solve the variable coefficient convection-diffusion equation with high-Reynolds number. Scaled inter-grid transfer operators and potential on vectorization and parallelization are discussed. The high-order multigrid method is unconditionally stable and produces solution of 4th-order accuracy. Numerical experiments are included.

Key words: Multigrid method, high-order discretization, scaled residual transfer operator, convection-diffusion equation.

1 Introduction

Numerical simulation of the convection-diffusion equation plays a very important role in modern large scale scientific computation, especially in computational fluid dynamics. The general convection-diffusion equation with Dirichlet boundary conditions is of the form

$$\left.\begin{aligned} u_{xx}(x,y)+u_{yy}(x,y)+p(x,y)u_x(x,y)+q(x,y)u_y(x,y) &= f(x,y), && (x,y)\in\Omega, \\ u(x,y) &= g(x,y), && (x,y)\in\partial\Omega, \end{aligned}\right\} \quad (1)$$

where $p(x,y)$ and $q(x,y)$ are functions of x and y, and simulate the Reynolds number in the case of the viscous flow problems. Ω is a convex domain.

When (1) is discretized by some difference formula, it results a system of linear equations

$$A^h u^h = f^h, \quad (2)$$

where the superscript h indicates the uniform mesh-size and the coefficient matrix A^h is nonsymmetric and not positive definite in general. To facilitate discussion, we define the cell Reynolds number associated with the mesh-size h as

$$\mathrm{Re} = \max\Big(\sup_{(x,y)\in\Omega}|p(x,y)|, \sup_{(x,y)\in\Omega}|q(x,y)|\Big)\cdot h/2.$$

For $\mathrm{Re}\le 1$, we say that equation (1) is diffusion-dominated. Otherwise it is convection-dominated. In this paper, we are primarily interested in the case when $\mathrm{Re}\to\infty$.

*Submitted for the Proceedings of the 1996 Copper Mountain Conference on Iterative Methods.
†Email address: zhang@math.gwu.edu.

If the discretization is the central differences, the resulting scheme is a five-point formula ($\mathcal{FPF}$) and has a truncation error of order h^2 and iterative methods for solving the resulting system of linear equations do not converge when Re is greater than a certain constant. Although the upwind scheme is unconditionally stable it has a truncation error of order h.

Since A^h is in general nonsymmetric, finding a stable numerical solution of (2) for large Re is one of the hardest and hottest problems in multigrid. de Zeeuw [12] developed a black-box multigrid solver with some matrix-dependent prolongations and restrictions. Acceleration techniques based on over-weighted residual transferring and defect correction were proposed by Brandt and Yavneh [2]. A cyclic reduction preconditioner was used with multigrid by Golub and Tuminaro [3]. A multigrid method based on Schur complement of the coefficient matrix and the matrix-dependent prolongation operator was recently published by Reusken [9]. Most of these multigrid methods are based on some kinds of $\mathcal{FPF}$s, but their preconditioned systems are usually some kinds of nine-point formulas ($\mathcal{NPF}$s).

Multigrid applications of $\mathcal{NPF}$ schemes have been proposed for special form of (1), i.e. the Poisson equation [10] and the constant coefficient case [6]. In this paper we employ a compact $\mathcal{NPF}$ with multigrid techniques to solve equation (1) with variable coefficients for large Re and investigate the potential on the parallelization and vectorization of the multigrid with $\mathcal{NPF}$.

This paper is organized as follows: In § 2, we present the high-order multigrid method and compare some existing approaches. In § 3, we discuss the issue of scaling the inter-grid operators. In § 4, several test problems are solved by the proposed method to show the stability and the effectiveness of the $\mathcal{NPF}$ multigrid solver for large Re. Conclusions and remarks are given in § 5.

2 High-Order Multigrid Method

The approximate value of a function $u(x,y)$ at a mesh point (x,y) is denoted by u_0. Those at its eight immediate neighboring points are denoted by $u_i, i = 1,2,\ldots,8$. The discretized values of p_i, q_i and $f_i, i = 0,1,\ldots,4$, have their obvious meanings. The finite difference formula for the mesh point (x,y) involves the nearest eight neighboring mesh points with mesh-size h and has the following computational stencil (the Southwell notation):

$$\begin{pmatrix} u_6 & u_2 & u_5 \\ u_3 & u_0 & u_1 \\ u_7 & u_4 & u_8 \end{pmatrix} \tag{3}$$

The high order finite difference formula for equation (1) is given by (see [4] for details):

$$\sum_{j=0}^{8} \alpha_j u_j = \frac{h^2}{2}[8f_0 + f_1 + f_2 + f_3 + f_4] + \frac{h^3}{4}[p_0(f_1 - f_3) + q_0(f_2 - f_4)], \tag{4}$$

where h is the uniform mesh-size. The coefficients $\alpha_i, i = 0,1\ldots,8$, are

$$\begin{aligned}
\alpha_0 &= -[20 + h^2(p_0^2 + q_0^2) + h(p_1 - p_3) + h(q_2 - q_4)], \\
\alpha_1 &= 4 + \frac{h}{4}[4p_0 + 3p_1 - p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 + p_0(p_1 - p_3) + q_0(p_2 - p_4)], \\
\alpha_2 &= 4 + \frac{h}{4}[4q_0 + 3q_2 - q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 + p_0(q_1 - q_3) + q_0(q_2 - q_4)],
\end{aligned}$$

$$
\begin{aligned}
\alpha_3 &= 4 - \frac{h}{4}[4p_0 - p_1 + 3p_3 + p_2 + p_4] + \frac{h^2}{8}[4p_0^2 - p_0(p_1 - p_3) - q_0(p_2 - p_4)], \\
\alpha_4 &= 4 - \frac{h}{4}[4q_0 - q_2 + 3q_4 + q_1 + q_3] + \frac{h^2}{8}[4q_0^2 - p_0(q_1 - q_3) - q_0(q_2 - q_4)], \\
\alpha_5 &= 1 + \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0q_0, \\
\alpha_6 &= 1 - \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0q_0, \\
\alpha_7 &= 1 - \frac{h}{2}(p_0 + q_0) + \frac{h}{8}(q_1 - q_3 + p_2 - p_4) + \frac{h^2}{4}p_0q_0, \\
\alpha_8 &= 1 + \frac{h}{2}(p_0 - q_0) - \frac{h}{8}(q_1 - q_3 + p_2 - p_4) - \frac{h^2}{4}p_0q_0.
\end{aligned}
$$

Numerical experiments reported in [4] show that this scheme converges for any values of $p(x, y)$ and $q(x, y)$ with SOR method.

When Re $\equiv 0$, equation (1) reduces to the Poisson equation, scheme (4) reduces to the well-known Mehrstellen formula. Multigrid applications of Mehrstellen formula have been investigated by Schaffer [10], Gupta, Kouatchou and Zhang [5]. When $p(x, y)$ and $q(x, y)$ are constants we [6] have shown that $\mathcal{NPF}$ with multigrid converges for any Re and demonstrates 4th-order accuracy while $\mathcal{FPF}$ diverges for some Re > 1 and demonstrates 2nd-order accuracy for small Re.

Formula (4) is compact in the sense that updating value at one point uses values at its eight nearest neighboring points. No special formula is needed for points near the boundary.

Algorithm 2.1 describes a standard recursive multigrid cycling scheme, where R and P are the restriction and interpolation operators respectively. In practice, μ is chosen to be 1 or 2 and the resulting multigrid cycling schemes are called the V-cycle and the W-cycle schemes respectively.

Algorithm 2.1 *Multigrid Cycling Scheme*

$v^h \leftarrow MG(v^h, f^h)$

Step 0: *If Ω^h = the coarsest grid, solve $v^h = (v^h)^{-1} f^h$, otherwise,*

Step 1: *Relax ν_1 times on $A^h v^h = f^h$ with a given initial guess v^h.*

Step 2(a): $r^h \leftarrow f^h - A^h v^h$.

Step 2(b): $r^{2h} \leftarrow R r^h$.

Step 2(c): $f^{2h} \leftarrow r^{2h}$.

Step 2(d): $v^{2h} \leftarrow 0$.

Step 2(e): $v^{2h} \leftarrow MG(v^{2h}, f^{2h})$ *μ times.*

Step 3: *Correct* $v^h \leftarrow v^h + P v^{2h}$.

Step 4: *Relax ν_2 times on $A^h v^h = f^h$ with the initial guess v^h.*

The discretized grid space is naturally (lexicographically) ordered. For $\mathcal{NPF}$ we may re-order the grid space by systematically coloring all grid points with four colors so that relaxation on grid points with different colors can be carried out simultaneously and independently [1]. In this paper, we investigate the four-color ordering of the grid space with Gauss-Seidel relaxation method. The bi-linear interpolation will be used in our algorithm to interpolate the coarse-grid-correction (CGC) to the fine grid. The scaled residual-injection operator discussed below will be used to transfer the residuals from the fine grid to the coarse grid (see [8]).

The compact $\mathcal{NPF}$ scheme (4) in conjunction with the four-color Gauss-Seidel relaxation is used with multigrid technique and the resulting method is referred to as $\mathcal{NPF}$-MG. The same $\mathcal{NPF}$ is used on all grids which greatly simplifies the coding.

We close this section by comparing our $\mathcal{NPF}$-MG with some existing multigrid methods mentioned in § 1. Golub and Tuminaro's cyclically preconditioned method [3] produces a non-compact nine-point stencil. Special treatments are needed for unknowns near the boundary. For coarse grid operators, techniques are employed to avoid the increase of the number of unknowns in the stencil. Also, artificial dissipation terms are added when the coarse grid operator is not diagonally dominated. The same difficulty was experienced by Reusken [9], who had to use the "lumping method" to approximate $\mathcal{NPF}$ by $\mathcal{FPF}$ on the coarse grid. Also, Reusken assumed that the matrix with $\mathcal{NPF}$ is weakly diagonally dominated so that the approximating matrix with $\mathcal{FPF}$ is an M-matrix to guarantee convergence. The diagonal dominance is usually guaranteed by the upwind discretization, as used by Reusken in [9], the upwind scheme renders the resulting method 1st-order accuracy. Although it has not been proved, our numerical experiments showed that weakly diagonal dominance is not necessary to guarantee convergence for $\mathcal{NPF}$-MG. These existing methods suggest that $\mathcal{NPF}$ be beneficial for stability. Our $\mathcal{NPF}$-MG uses $\mathcal{NPF}$ directly without any preconditioner or added artificial dissipation terms.

3 Scaling Residual Transfer Operators

The efficiency of a multigrid algorithm depends on the quality of the relaxation method (the smoother) and on the quality of CGC, which is to correct the current approximation by solving a sub-problem on a coarser grid. In this paper, we are interested in optimizing CGC process, the issue of how to find a good smoother will not be discussed here.

Let Ω^h be the grid space associated with mesh-size h and Ω^{2h} with $2h$. Let there be a regular splitting of A^h, M and N be nonsingular square matrices satisfying the consistency condition

$$M + NA^h = I.$$

Given $v^h \in \Omega^h$, denote the error $e^h = u^h - v^h$, where $u^h = (A^h)^{-1} f^h$ is the exact solution. The nature of CGC is to look for a $v^{2h} \in \Omega^{2h}$ to fulfill the minimization condition (see [11])

$$\|e^h - Pv^{2h}\|_Z = \min_{w \in \Omega^{2h}} \|e^h - Pw\|_Z, \tag{5}$$

where $\|\cdot\|_Z$ is the energy norm with respect to some symmetric, positive definite matrix Z on Ω^h.

As Brandt and Yavneh [2] remarked that, when solving the convection-diffusion equations with large Re, the error is dominated by smooth components. Hence, instead of increasing smoothing sweeps on the fine grid, they concentrated their efforts on improving the CGC process because in many cases, the coarse grid solution fails to approximate that of the fine grid.

In many practical applications, especially when A^h is not symmetric positive definite (NSPD), satisfying (5) is not enough to guarantee fast convergence. Modifications of standard multigrid CGC have been proposed (see [2, 7, 9, 11, 13, 15]). At the beginning of each multigrid cycle (see Algorithm 2.1), suppose e_0 is the error after Step 1, and e_1 is the error after Step 3, in this paper, we concentrate on fulfilling the following minimization problem

$$\|e_1\|_Z / \|e_0\|_Z = \min_{\alpha \in R} \|u^h - (v^h + \alpha P v^{2h})\|_Z / \|e_0\|_Z. \tag{6}$$

Unfortunately, (6) is not directly solvable, because the error e_0 and the exact solution u^h are not known. Hence, we need some way to estimate the optimal α.

Suppose α can be estimated by some suitable method we modify Algorithm 2.1 by

The 1st approach: Step 3: $v^h = v^h + \alpha P v^{2h}$,

which will be referred to as the *post-scaling technique,* or by

The 2nd approach: Step (2b): $r^{2h} = \beta R r^h$,

which will be referred to as the *pre-scaling technique.* We have unified these (and other) approaches as the *residual scaling techniques* in [14]. The following theorem was proved in [14]:

Theorem 3.1 *The pre-scaling and the post-scaling techniques are mathematically equivalent if and only if the scaling parameters are equal, i.e. if and only if* $\alpha = \beta$.

Due to Theorem 3.1 we will no longer distinguish the scaling parameters α and β, but just use α to denote both the pre-scaling and the post-scaling parameters.

Now, we discuss how to compute or estimate the scaling parameter α in (6). In [11], Vaněk showed that, if we choose $Z = A^h$, (6) is equivalent to the following minimization problem

$$\|M^{\nu_2}[e^h - \alpha P v^{2h}]\|_{A^h} = \min_{\bar{\alpha} \in \mathcal{R}} \|M^{\nu_2}[e^h - \bar{\alpha} P v^{2h}]\|_{A^h}. \tag{7}$$

This approach (termed as over-correction by Vaněk in [11]) computes α in the process of CGC and requires that A^h be symmetric and positive definite (SPD), which is not the case for equation (1) with large Re. The SPD requirement puts a severe limit on the application of many post-scaling acceleration techniques. In addition, one step of the over-correction usually costs more than one multigrid cycle and a large number of pre-smoothing sweeps is needed. In Vaněk's test problem for solving an anisotropic Poisson equation [11], he used 7 pre-smoothing and 2 post-smoothing sweeps (see [11]). The cost is prohibitive.

Reusken's approach [9] also belongs to the 1st approach, but no specific method is proposed to estimate α. Although Reusken included some impressive numerical results show that the contraction numbers are very small for his algorithm, his method was derived from the upwind scheme and thus suffers from the same drawback as being of 1st-order accuracy.

In [2], Brandt and Yavneh took the 2nd approach and used a heuristic analysis to estimate α for solving high-Re flows. An over-weighted residual method ($\alpha = \sqrt{2}$) and defect correction technique were proposed. It was shown that the convergence of the multigrid solver based on the first differential approximation is considerably improved.

In [7], We also took the 2nd approach and used a heuristic residual analysis based on the geometry of the grid points to find an optimal α (= 0.5424) for $\mathcal{NPF}$-MG with red-black ordering for the diffusion-dominated problems. It has been shown [7] that this α improves the cost-effectiveness of $\mathcal{NPF}$-MG when diffusion dominates. It also guarantees convergence when convection dominates, while full-weighting is divergent.

Except for the cost of estimating or computing α, the implementation of the 2nd approach is cost-free, but the implementation of the 1st approach is one multiplication on the fine grid.

Clearly, if the residuals need to be scaled before they are restricted to the coarse grid, there may be no big difference to scale the residuals from full-weighting or from injection (with different scaling factor). Hackbusch [8] remarked that injection is computationally optimal and we may save 3/4 of the residual restriction cost by scaling residuals from injection,

In parallel implementation, injection is clearly advantageous over full-weighting because injection is a local process and requires no communication with neighboring processors. Full-weighting requires communication with eight neighboring grid point which may be in different processors.

Moreover, when $p(x,y)$ and $q(x,y)$ are oscillatory on Ω, the direction of the convection changes rapidly. In particular, when Ω contains turning-points, the convection changes direction at the turning-points and equation (1) represents a recirculating flow. The full-weighting operator usually mis-represents the characteristics of the flow around the turning-points. By projecting residuals with mis-represented characteristics to the coarse-grid, the coarse-grid sub-problem fails to approximate that of the fine-grid at all and causes divergence on the find-grid for high-Re recirculating flow. On the other hand, injection may maintain the characteristics in this case.

Although the order of the injection operator is 0 (see [8]) and the combination of injection with bi-linear interpolation (order 2) violates the order rule set up by Brandt and Hackbusch [8] for small Re (2nd-order equation), our numerical experiments showed that the CPU cost is almost the same by using scaled injection and by using full-weighting (the convergence deteriorates slightly). For large Re, equation (1) approaches a 1st-order equation and the combination of our inter-grid transfer operators satisfies the order rule.

Hence, we choose the 2nd approach and use the scaled injection operator with a scaling parameter α in our $\mathcal{NPF}$-MG solver. For the four-color Gauss-Seidel relaxation and for solving equation (1) with large Re, we have found that $\alpha = 5$ yields very good numerical results. Note that our $\mathcal{NPF}$-MG gives same results (convergence and computed accuracy) for all Re $\geq 10^3$. For small Re, α changes as Re increases or decreases in some range, we recommend that full-weighting be used as the restriction operator. However, since $\mathcal{NPF}$-MG with full-weighting diverges for problems with turning points and large Re, the scaled injection operator should be used if the magnitude of the convection coefficients is not known *a priori*. Like the development of general numerical softwares, there is a trade-off between the robustness and the efficiency.

4 Numerical Experiments

For numerical experiments, we solve equation (1) with the convection coefficients as follows:

Test Problem 1:	$p(x,y) = K,$	$q(x,y) = K;$
Test Problem 2:	$p(x,y) = Kx(2y-1),$	$q(x,y) = Ky^2(5-3x);$
Test Problem 3:	$p(x,y) = Kx\cos(x+y),$	$q(x,y) = Ky\sin(x+y);$
Test Problem 4:	$p(x,y) = K\exp(x+y),$	$q(x,y) = K\exp(-x-y).$

We choose $\Omega = (0,1)\times(0,1)$. Note that Problems 2 and 3 have turning-points in Ω and $\mathcal{NPF}$-MG with full-weighting diverges for large Re. All problems were solved for the same conditions and by using a uniform mesh-size h. The boundary values were given so that the exact solutions were $u(x,y) = x^2+y^2$. The initial guesses were $u(x,y) = 0$ for all problems. Since it has been established (see [6, 7]) that our $\mathcal{NPF}$-MG yields numerical solution of 4th order accuracy for equation (1), we do not report the computed accuracy here.

The computations were done on a vector machine Cray-90 at the Pittsburgh Supercomputing Center. The program was coded in Fortran 77 language and compiled by Cray Fortran 77 compiler in single precision (roughly 16 digits of accuracy).

In Algorithm 2.1, $\mu = 2$ was chosen (W-cycle). On the finest grid, the mesh-sizes were chosen as $h = 1/32, 1/64, 1/128,$ and $1/256$. Standard coarsening technique was used and the coarsest grid has a mesh size $h = 1/2$.

We solved equation (1) for large Re by choosing K so that $\mathrm{Re} \in [10^3, 10^{300}]$. Re can really approach infinity, the limit is on computer's hardware. $\alpha = 5$ was chosen and two pre-smoothing and two post-smoothing sweeps ($\nu_1 = \nu_2 = 2$) were applied on all grids.

The convergence histories of the four test problems with different finest mesh-sizes are depicted in Figure 1. The data reported were obtained by choosing $K = 10^{100}$. We point out that exactly the same picture may be obtained for all $\mathrm{Re} \in [10^3, 10^{300}]$. We conclude that the convergence rate of $\mathcal{NPF}$-MG is not affected by Re for large Re. Judging from Figure 1, we can find that $\mathcal{NPF}$-MG converges satisfactorily. We point out, except for Test Problem 1, the computed solutions almost reach the limit of the algorithm.

5 Conclusions and Remarks

A compact nine-point discretization formula has been used with the multigrid technique to develop a stable and high accuracy multigrid solver ($\mathcal{NPF}$-MG) for the variable coefficient convection-diffusion equation with large Re. Techniques for improving CGC by scaling the residual transfer operators are discussed. Vectorization and parallelization potentials of $\mathcal{NPF}$-MG are investigated and the solver was tested on a vector machine with the four-color Gauss-Seidel relaxation. Four test problems were solved to demonstrate the efficiency of our solver. Numerical experiments show that the convergence rate of $\mathcal{NPF}$-MG is not affected by the magnitude of Re beyond some constant.

The beauty of $\mathcal{NPF}$-MG is that it requires no preconditioner nor added dissipation terms for high-Re problems. The coding of the program is simple. Our $\mathcal{NPF}$-MG solver was developed by modifying a standard $\mathcal{FPF}$ Poisson solver. With four-color Gauss-Seidel relaxation method and the properly scaled residual injection operator, $\mathcal{NPF}$-MG can be fully vectorized and parallelized.

We remark that the convergence rate of $\mathcal{NPF}$-MG can be accelerated if we replace the four-color Gauss-Seidel relaxation by the four-color SOR relaxation [1]. The suitability of vectorization and parallelization will not change. If one is interested in solving problems on serial computers we recommend the red-black $\mathcal{NPF}$-MG in [7] which has better convergence. However, for parallel computation the current implementation seems better. Other multigrid acceleration schemes such as the *optimal residual scaling techniques* and the *minimal residual smoothing techniques* have been investigated intensively by us and are reported in [14, 15, 16] (up to 88% acceleration in convergence).

References

[1] L. Adams and J. Ortega, A multi-color SOR method for parallel computation, *Proc. of 1982 Int. Conf. on Paral. Processing*, (K.E. Batcher *et al*, eds), 53–56 (1982).

[2] A. Brandt and I. Yavneh, Accelerated multigrid convergence and high-Reynolds recirculating flows, *SIAM J. Sci. Comput.* 14, 607–626 (1993).

[3] G. H. Golub and R. S. Tuminaro, Cyclic reduction/multigrid, Tech. Rep. NA-92-14, Stanford University, Stanford, CA, (1992).

[4] M. M. Gupta, R.P. Manohar, and J.W. Stephenson, A single cell high order scheme for the convection-diffusion equation with variable coefficients, *Int. J. Numer. Methods Fluid.* 4, 641–651 (1984).

[5] M.M. Gupta, J. Kouatchou and J. Zhang, Comparison of 2nd and 4th order discretizations for multigrid Poisson solver, *J. Comput. Phys.* (submitted, 1995).

[6] M.M. Gupta, J. Kouatchou and J. Zhang, An accurate and stable multigrid method for convection-diffusion equation, (submitted, 1995).

[7] M.M. Gupta, J. Kouatchou and J. Zhang, Preconditioning free multigrid method for convection-diffusion equation with variable coefficients, (submitted, 1995).

[8] W. Hackbusch, *Multi-grid Methods and Applications,* 1985.

[9] A. Reusken, Fourier analysis of a robust multigrid method for convection-diffusion equations, *Numer. Math.* **71**, 365–397 (1995).

[10] S. Schaffer, High order multi-grid methods, *Math. Comput.* **43**, 89–115 (1984).

[11] P. Vaněk, Fast multigrid solver, *Appl. of Math.* **40**, 1–20 (1995).

[12] P.M. de Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, *J. Comput. Appl. Math.* **33**, 1–27 (1990).

[13] J. Zhang, Acceleration of five-point Red-Black Gauss-Seidel in multigrid for two dimensional Poisson equation, *Appl. Math. Comput.* (to appear).

[14] J. Zhang, Residual scaling techniques in multigrid, I: equivalence proof, (submitted, 1996).

[15] J. Zhang, Minimal residual smoothing in multi-level iterative method, *Appl. Math. Comput.* (to appear).

[16] J. Zhang, Analysis of minimal residual smoothing in multigrid, (submitted, 1996).

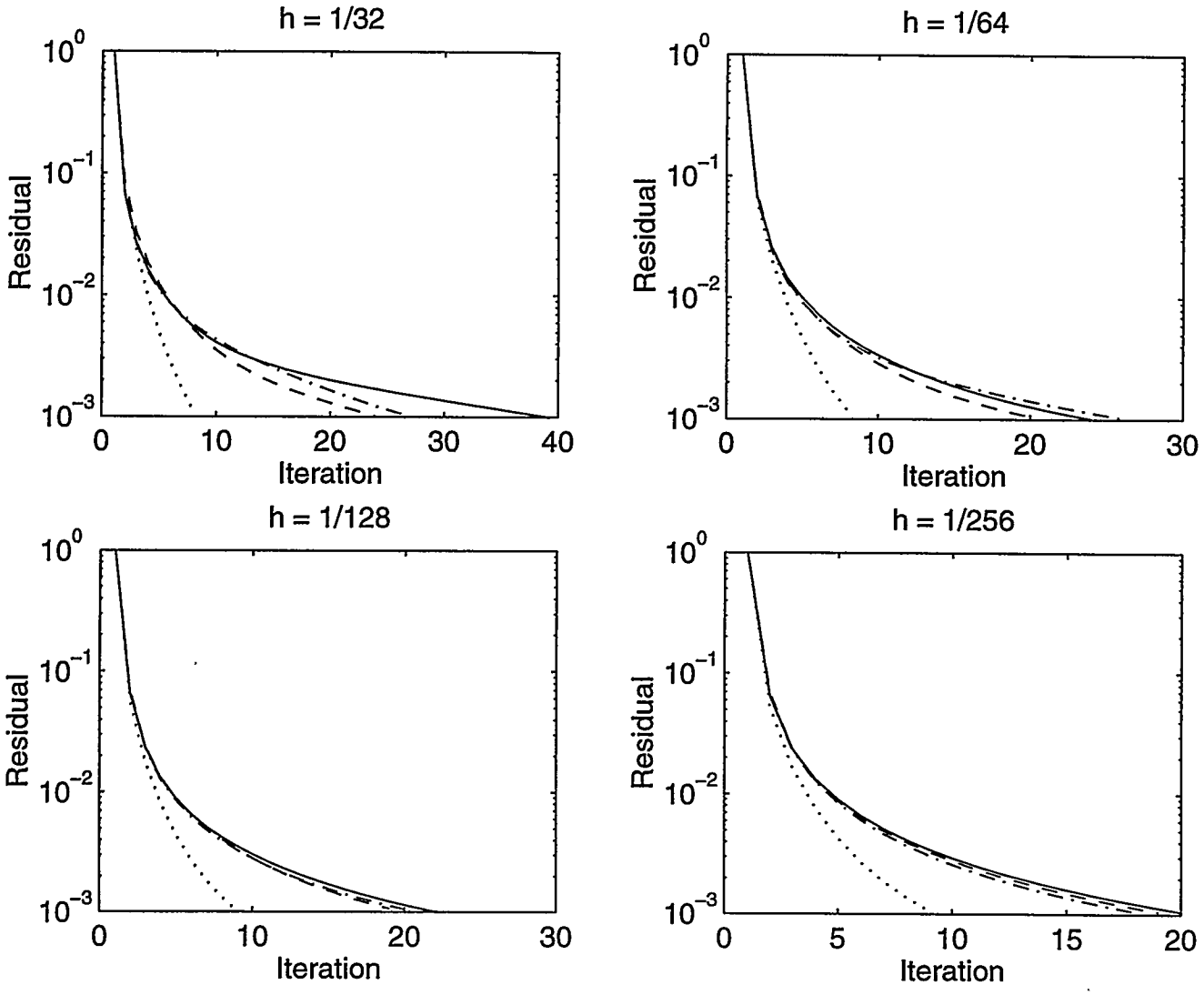


Figure 1: The convergence histories (in log scale) with $K = 10^{100}$ and different h. Dotted line is for Test Problem 1, solid line for Test Problem 2, dot-dashed line for Test Problem 3, dashed line for Test Problem 4.

Topic: *Markov Chains*	*Session Chair:* *Daniel Szyld*	*Room C*
4:45 - 5:15	T. Dayar	State Space Orderings for Gauss-Seidel in Markov Chains Revisited
5:15 - 5:45	G. Horton	On the Multi-Level Solution Algorithm for Markov Chains
5:45 - 6:15	D. Szyld	Threshold Partitioning of Sparse Matrices and Applications to Markov Chains
6:15 - 6:45		

STATE SPACE ORDERINGS FOR GAUSS–SEIDEL IN MARKOV CHAINS REVISITED

TUĞRUL DAYAR[1]

Abstract. Symmetric state space orderings of a Markov chain may be used to reduce the magnitude of the subdominant eigenvalue of the (Gauss–Seidel) iteration matrix. Orderings that maximize the elemental mass or the number of nonzero elements in the dominant term of the Gauss–Seidel splitting (that is, the term approximating the coefficient matrix) do not necessarily converge faster. An ordering of a Markov chain that satisfies Property–R is semi–convergent. On the other hand, there are semi–convergent symmetric state space orderings that do not satisfy Property–R. For a given ordering, a simple approach for checking Property–R is shown. An algorithm that orders the states of a Markov chain so as to increase the likelihood of satisfying Property–R is presented. The computational complexity of the ordering algorithm is less than that of a single Gauss–Seidel iteration (for sparse matrices). In doing all this, the aim is to gain an insight for faster converging orderings. Results from a variety of applications improve the confidence in the algorithm.

Key words. State space ordering, Markov chains, Gauss–Seidel, Property–R

AMS subject classifications. 65U05, 60J10, 60J27, 65F10, 65F50, 65F30, 65B99

References

[1] G. R. GILBERT, *Predicting structure in sparse matrix computations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 62–79.

[2] L. KAUFMAN, *Matrix methods for queueing problems*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 525–552.

[3] D. MITRA AND P. TSOUCAS, *Relaxations for the numerical solutions of some stochastic problems*, Communications in Statistics: Stochastic Models, 4 (1988), pp. 387–419.

[4] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, New Jersey, 1994.

[1]Department of Computer Engineering and Information Science, Bilkent University, 06533 Bilkent, Ankara, Turkey (tugrul@bilkent.edu.tr).

On the Multi-Level Solution Algorithm for Markov Chains

Graham Horton [1]
Computer Science Department,
University of Erlangen-Nürnberg,
Martensstr. 3, 91058 Erlangen, Germany.

Abstract

We discuss the recently introduced multi-level algorithm for the steady-state solution of Markov chains. The method is based on the aggregation principle, which is well established in the literature. Recursive application of the aggregation yields a multi-level method which has been shown experimentally to give results significantly faster than the methods currently in use. The algorithm can be reformulated as an algebraic multigrid scheme of Galerkin-full approximation type. The uniqueness of the scheme stems from its solution-dependent prolongation operator which permits significant computational savings in the evaluation of certain terms.

1 Introduction

Markov chains describe discrete-state stochastic processes in which the probabilities of transitions between states are a function solely of the current state of the chain - the so-called memoryless property. Since this property is approximately satisfied by many physical systems, Markov chains are used widely in stochastic modeling. We will draw our examples in this paper from the field of computer performance modeling. It is common to distinguish between continuous time Markov chains (CTMCs), in which transition coefficients between states are interpreted as exponentially distributed rates or delays, and discrete time Markov chains (DTMCs), where they are treated as probabilities. In the latter case, the Markov chain is described by a stochastic matrix. Since, however, for the steady-state case, CTMC problems can be converted via a simple transformation into problems described by a DTMC, we will henceforth restrict ourselves to the latter. Ultimately, the Markov chain represents a linear system of equations which is usually very sparse and often extremely large.

[1]Email: *graham@informatik.uni-erlangen.de.* This work was carried out while the author was a guest at ICASE, NASA Langley Research Center, Hampton, VA. and a visitor at the Mathematics and Computer Science Department of the University of Denver.

One goal of modeling computer systems is to derive information on performance, measured typically as job throughput or component utilization, and availability, defined as the proportion of time a system is able to perform a certain function in the presence of component failures and possibly also repairs. Various abstract modeling tools for computer systems are in widespread use today, the most important of which are generalized stochastic Petri nets (GSPNs) [1] and queueing networks [6]. When the memoryless condition is satisfied, such models are equivalent to Markov chains, and it is required to solve the Markov chain in order to derive useful information about the abstract model.

Unfortunately, the number of states of the Markov chain (and thus the dimension of the linear system) grows extremely quickly as the complexity of the model is increased. There is one unknown for each state that the model may be in - a number that is subject to a combinatorial explosion. Thus the Markov chains that have to be solved even for relatively coarse computer models may have tens or hundreds of thousands of states. Apart from their size, one further drawback of typical Markov chains is the presence of coefficients on a wide range of scales. Consider, for example, a reliability model of a computer, in which the rate of component failure may be only once in every few months, whereas the rates associated with the normal behaviour of the system are measured in kHz and MHz.

The resulting large systems of equations must be solved numerically using an iterative scheme. Typical iterative methods in use in the computer modelling community are the Power, Gauss-Seidel (GS), and successive over-relaxation (SOR) algorithms. Surveys of currently used methods may be found in [12, 8]. All of these methods have the drawback that they may require many iterations to reach an accurate solution, particularly if the system is large or if coefficients of strongly varying magnitude are present. This can lead to unacceptably long computation times.

In this paper we will consider the multi-level (ML) solution algorithm for Markov chains, which was introduced in [4]. The method is based on the principle of iterative aggregation and disaggregation, a well-established numerical solution technique for Markov chains [7, 15, 14]. It is shown that the method is equivalent to a algebraic multigrid scheme which uses the Galerkin method for the coarse level operator and is of Full Approximation scheme (FAS) type. The novelty of the method stems from the definition of the prolongation operator, which is solution-dependent and commutes from the left and from the right with the restriction operator. This has two interesting effects: the right hand side of the coarse level equations degenerates into a simple restriction of the fine-level right hand side, and the coarse level operator is solution-dependent and therefore changes from iteration to iteration, even though the original problem is linear.

In the following section we describe the problem and the aggregation equations. Then the multi-level method is described. In section 4 the multi-level method is rewritten as a multigrid scheme. In section 5 experimental results for Markov chains arising from a well-known multiprocessor reliability model and from a simple queueing network are presented, showing the superiority of the method over the standard iterations. In the final section we summarize the paper.

THRESHOLD PARTITIONING OF SPARSE MATRICES AND APPLICATIONS TO MARKOV CHAINS

HWAJEONG CHOI* AND DANIEL B. SZYLD*

Abstract. It is well known that the order of the variables and equations of a large, sparse linear system influences the performance of classical iterative methods. In particular if, after a symmetric permutation, the blocks in the diagonal have more nonzeros, classical block methods have a faster asymptotic rate of convergence. In this paper, different ordering and partitioning algorithms for sparse matrices are presented. They are modifications of PABLO [*SIAM J. Sci. Stat. Computing* **11** (1990) 811–823]. In the new algorithms, in addition to the location of the nonzeros, the values of the entries are taken into account. The matrix resulting after the symmetric permutation has dense blocks along the diagonal, and small entries in the off-diagonal blocks. Parameters can be easily adjusted to obtain, for example, denser blocks, or blocks with elements of larger magnitude. In particular, when the matrices represent Markov chains, the permuted matrices are well suited for block iterative methods that find the corresponding probability distribution. Applications to three types of methods are explored: (1) Classical block methods, such as Block Gauss Seidel. (2) Preconditioned GMRES, where a block diagonal preconditioner is used. (3) Iterative aggregation method (also called aggregation/disaggregation) where the partition obtained from the ordering algorithm with certain parameters is used as an aggregation scheme. In all three cases, experiments are presented which illustrate the performance of the methods with the new orderings. The complexity of the new algorithms is linear in the number of nonzeros and the order of of the matrix, and thus adding little computational effort to the overall solution.

1. Introduction.

O'Neil and Szyld [10] present an algorithm called PABLO that produces a symmetric permutation of a sparse matrix with the goal of obtaining dense diagonal blocks. They show that their algorithm is linear in the number of nonzeros and the order of the corresponding matrix. Their algorithm is combinatorial in nature, i.e., it takes into account the zero-nonzero structure or graph of the matrix, but not the values of its entries. Dutto et. al. [6] use a modified version of PABLO, where the number of blocks in the diagonal is prescribed and where they are of roughly the same size, to produce effective block parallel preconditioners for the solution of Navier-Stokes equations. Recently, similar ideas have been used for the parallel solution of nonlinear systems [11], [22].

In this contribution, we present some modified versions of PABLO, where, in addition to the combinatorial aspects, the values of the entries are taken into account. Thus, a threshold is introduced, and at the same time that a node in the graph of the matrix is tested for inclusion in the component corresponding to a diagonal block, the value of

* Department of Mathematics, Temple University, Philadelphia, Pennsylvania 19122-2585, USA (choi@math.temple.edu, szyld@math.temple.edu). This work was supported by the National Science Foundation grant DMS-9201728.

its entry (or of other entries) is compared with the given threshold; see Section 2. The ordering and partitions produced by the new algorithms depend on the threshold and other connectivity parameters. By changing the parameters, different blocks can be obtained, e.g., with entries within the diagonal blocks that are larger in magnitude. Since the new algorithms have an additional search through several nodes, their complexity is higher than that of PABLO, but it is still linear in the number of nonzeros and the order of the matrix, as shown in [3]. This is also illustrated in the experiments in Section 3.

The new algorithms can be applied to any square matrix, and more specifically to singular matrices. The applications we have in mind include finding the stationary probability distribution of a Markov Chain; see e.g., [17]. In the survey [12], several iterative methods for the solution of such systems are discussed. We have chosen three types of iterative methods, and applied the partitions obtained with the original PABLO and the new threshold algorithms. The methods considered are classical block iterative methods such as Block Gauss Seidel [21], preconditioned GMRES [13] with block diagonal preconditioning, and aggregation/disaggregation methods; see, e.g., [2], [9], [14], [19].

We have found first, that the ordering generated by the original PABLO is a good partition for the matrices corresponding to these problems. Furthermore, the orderings produced by the new algorithms, are obtained in similar time, and generally give rise to even faster convergence; see Section 3.

2. Threshold Orderings. In this section we briefly review the algorithm PABLO (PArametrized BLock Ordering) and describe the threshold variants. Given an $n \times n$ matrix $A = (a_{ij})$, let $G = (V, E)$ be its associated graph, i.e., $V = \{v_1, \ldots, v_n\}$ is the set of n vertices and E is the set of edges, where $(v_i, v_j) \in E$ if and only if $a_{ij} \neq 0$; see, e.g., [5], [7]. Given this graph, PABLO constructs q subgraphs $G_k = (V_k, E_k)$, $k = 1, \ldots, q$. The number of subgraphs, q, i.e., the number of the corresponding diagonal blocks, is not known a priori, but is determined by the algorithm and it depends on the structure of the graph G and the input parameters. In the algorithm PABLO (as well as in the threshold algorithms described in this section), a first node is taken from the queue of unmarked nodes and this node starts a new current set of vertices P, then additional nodes are taken from the queue and added to the set P if they satisfy certain criteria, or sent back to the queue if not. Two criteria are used in PABLO to determine if a node v should be added to a current set $P \subset V$, corresponding to a diagonal block, by measuring how full $v \cup P$ is, and to what degree the vertices in $v \cup P$ are connected to each other. The first criterion is measured by the ratio of the total number of edges corresponding to $v \cup P$ to the number of edges that subgraph would have if it were complete (corresponding to a full submatrix). If it is satisfied then the new node is added. The second test is that the new node v must be adjacent to at least a certain proportion of nodes in the subgraph

corresponding to P and more than outside the subgraph. The two parameters which govern these criteria, referred as α and β, are recommended in [10] to have default values $\alpha = 1$, $\beta = 0.5$, or to be reset by the user. The linearity of the algorithm is obtained by selecting a set of eligible nodes, thus restricting the search to a relatively small set of nodes. For further details, see [10].

In the two variants of threshold PABLO (TPABLO) described here, a third additional criterion is used to decide if a new vertex v is added to the current subgraph being formed. Let γ be the given threshold, let P be the set of nodes of the current subgraph, and v_j be the vertex being tested for addition to P (corresponding to the jth row and column of the matrix). In each of the TPABLO versions, v_j is added to P if, in addition to either of the two criteria in PABLO, the following holds.
Criterion 1. $|a_{ij}| > \gamma$ or $|a_{ji}| > \gamma$ for at least one $i \in P$.
Criterion 2. $|a_{ij}| > \gamma$ and $|a_{ji}| > \gamma$ for all $i \in P$.

The use of criterion 1 produces a permuted matrix in which every entry in the off-diagonal blocks is smaller than the threshold in absolute value. This criterion is also useful to find NCD matrices; see, e.g., [1], [8], [18]. The use of criterion 2 produces a permuted matrix in which every entry in the diagonal block is larger than the threshold in absolute value, with the possibility that some entries in the off-diagonal blocks are also larger than the threshold in absolute value.

The following pseudo-code summarizes a portion of the algorithm TPABLO, cf. [10]. In it, the set $C \subset V$ is the set of vertices which have yet not been assigned to a block, the set $P \subset V$ is the current set, i.e., the one corresponding to a diagonal block being constructed, and the set $Q \subset V$ is the set of nodes in C which are adjacent to nodes in P. Thus, at the beginning of the algorithm, we have $C = V$, $P = Q = \emptyset$.

```
Given a set C, set q = 0
repeat until C = ∅
        let P = Q = ∅
        choose from C a node c, mark it, and place it on P
        move to Q all nodes in C adjacent to c
        repeat until Q = ∅
                chose the node p from the head of Q
                calculate connectivity information
                if either the fullness or the connectivity criteria of PABLO is satisfied
                and the threshold criterion holds (1 or 2, depending of the algorithm)
                then
                        mark p and move it to P
                        add to the rear of Q all nodes in C adjacent to p
                        update connectivity information
                else
```

move node p from Q to C
endif
designate those nodes in P to be in a block (and set $q = q + 1$)

The flexibility of PABLO and its threshold variants is increased by the introduction of an additional parameter *minbs* which guarantees a minimum size for each block along the diagonal. A value of $minbs = 0$ has no effect on the algorithm, while a very large value is obviously not recommended.

3. Numerical experiments. We present numerical experiments illustrating how the permutations generated by the two versions of TPABLO are useful to find stationary probability vectors of Markov chains. We begin by comparing the ordering and permutations obtained with these, to those generated by the algorithm provided in the package MARCA due to W. Stewart [16], [18]. We will call the latter algorithm MARCA for short. In it, given a threshold, all elements of magnitude below it are discarded, and the strongly connected components of the remaining graph are determined. We point out that the partition algorithm of MARCA, is essentially the same as the epsilon decomposition of Sezer and Šiljak [15] (using symmetric permutations). The matrices used in these experiments correspond to an interactive computer system, described as Example 1 in [12] or in [18]. The resulting NCD matrices corresponding to 20, 30 and 50 users have 1771, 5456 and 23426 states (number of variables) and 11,011, 35,216, and 156,026 nonzeros, respectively.

	order	γ	ngr	ω	it	Tp	Ta	Tb	Tt	rnorm
TPABLO1	1771	0.01	21	1.5	4	0.27	0.00	2.36	9.45	1.38e-09
MARCA		0.01	21	1.5	5	0.02	0.00	2.34	11.80	2.80e-09
TPABLO1	5456	0.05	496	1.3	4	0.55	2.32	0.28	10.82	5.07e-09
		0.01	31	1.5	4	1.75	0.00	17.06	68.41	5.06e-10
TPABLO2		0.05	496	1.3	4	0.57	2.36	0.27	10.97	5.07e-09
		0.01	31	1.5	4	1.75	0.00	17.06	68.41	5.06e-10
MARCA		0.05	496	1.3	5	0.05	2.39	0.28	13.88	2.53e-10
		0.01	31	1.5	5	0.12	0.00	17.43	87.40	8.57e-10
TPABLO1	23426	0.05	1326	1.3	7	3.25	18.03	2.37	147.06	2.31e-11
		0.01	51	1.4	5	20.45	0.03	206.25	1032.39	8.64e-11
MARCA		0.05	1326	1.3	6	0.37	17.93	2.32	125.53	2.17e-10
		0.01	51	1.4	6	0.55	0.03	213.84	1284.43	8.24e-11

TABLE 1
Aggregation/Disaggregation for NCD matrices, $\alpha = \beta = 0.5$, $minbs = 0$

In Table 1 and in the tables that follow it, the parameter γ is the threshold used in the MARCA algorithm and in the two versions of TPABLO, "ngr" is the number of

preconditioner	size of blocks	no. of blocks	k	no. of cycles	rnorm
None			10	20	1.17e-01
			15	20	1.77e-02
Natural order	10	171	10	20	4.90e-01
	50	36	10	20	6.59e+00
	100	18	10	20	6.98e-01
TPABLO1		21	10	20	1.19e-07
			15	3	1.52e-09

TABLE 2
GMRES(k) on an NCD matrix, $n = 1771$, $\alpha = \beta = 0.5$, $\gamma = 0.01$, $minbs = 0$

aggregation groups, i.e., the number of blocks along the diagonal after the permutation, ω is the relaxation parameter used in the SOR method for the solution of the linear system corresponding to each block if its order is larger than 50 (Gaussian elimination is used for the smaller blocks), and "it" is the number of aggregation and disaggregation steps needed for convergence. The times reported are CPU seconds of a SUN Microsystems Sparc 20 at the Department of Mathematics at Temple University. "Tp" is the time for the ordering algorithm, either MARCA or one of the versions of TPABLO, "Ta" is the average solution time for the aggregated matrix, "Tb" is the average solution time for the diagonal blocks, while "Tt" is the total CPU time for convergence. In the last column we present the Euclidean norm of the residual of the computed probability vector.

We observe that with the parameters $\alpha = \beta = 0.5$, $minbs = 0$, TPABLO1 and TPABLO2 obtain the same groups of states as MARCA does, but the order of the variables (states) within each block is different. This accounts for the difference in performance of the methods with the different partitions. In other words, the SOR method converges faster for the blocks in the diagonal (of order larger than 50) with the ordering produced by the TPABLOs. As it can be observed, for these NCD matrices, the times obtained with the TPABLOs are of the same order of magnitude, and in general better than those obtained with the MARCA partition. Different PABLO connectivity parameters give rise to different partitions which can produce better convergence time [3].

In Table 2 we present some preliminary results on experiments using GMRES(k) [13], i.e., with restarts every k GMRES iterations, using block diagonal preconditioners. We use the same NCD matrix as in Table 1 with 1771 states. For $k = 10$ restarts and 20 sets of restarted cycles (200 total GMRES iterations), runs with no preconditioning, as well as those with block diagonal preconditioning with blocks taken in the natural ordering, do not reduce the norm of the residual below 10^{-2}, while the blocks produced

by TPABLO give a very satisfactory answer. With $k = 15$ restarts, only 45 total GMRES iterations are needed for convergence.

	no. of blocks	no. of iter	total no. of operations	δ	Tp	Tc	rnorm
Point GS	1771	467	27927365	.9677		13.35	9.87e-09
2 by 2	886	467	29577664	.9567		13.24	9.87e-09
4 by 4	443	467	32878821	.9460		13.14	9.87e-09
PABLO	539	268	17350803	.9445	.09	7.30	9.79e-09
TPABLO1(0.05)	599	226	14150797	.9320	.11	5.96	9.86e-09
TPABLO1(0.01)	541	266	17268875	.9438	.11	7.24	9.61e-09
TPABLO1(0.001)	539	268	17350803	.9445	.09	7.08	9.79e-09
TPABLO2(0.05)	624	208	12847129	.9255	.13	5.88	9.53e-09

TABLE 3
Example 'CentralServer20', α=1.0, β=0.5

We turn now to numerical experiments using Block Gauss Seidel. We compare the results obtained with PABLO and its threshold variants with the standard (point) Gauss Seidel, and to block versions obtained by taking 2 by 2 and 4 by 4 blocks along the diagonal. In addition to CPU time, we report the total number of operations, as well as $\delta(T)$, the second largest eigenvalue of the corresponding iteration matrix. "Tc" is the total computing time, excluding the time for the partition algorithm. The minimum block size parameter was set to 2. The example in Table 3 is a matrix representing a standard queuing network systems described, e.g., in [20], with 1771 states and 9240 nonzeros. It was produced with the package SPNP [4]. It can be readily appreciated that the partitions generated by PABLO and the two versions of TPABLO provide better performance, combined with a block method, than point Gauss Seidel, as well as better than the 2 by 2 and 4 by 4 blocks.

Acknowledgements. We wish to thank Gianfranco Ciardo and William Stewart for generously providing us with the data used for our experiments.

REFERENCES

[1] Wei-Lu Cao and William J. Stewart. Iterative aggregation/disaggregation techniques for nearly uncoupled Markov chains. *Journal of the Association for Computing Machinery*, 32:702–719, 1985.

[2] Françoise Chatelin. Iterative aggregation / disaggregation methods. In G. Iazeolla, P. J. Courtois, and A. Hordijk, editors, *Mathematical Computer Performance and Reliability*, pages 199–207, Amsterdam – New York – Oxford, 1984. North-Holland.

[3] Hwajeong Choi and Daniel B. Szyld. Application of threshold partitioning of sparse matrices to markov chains. Research Report 96-21, Department of Mathematics, Temple University, Philadelphia, February 1996.

[4] Gianfranco Ciardo, Kishor S. Trivedi, and Jogesh Muppala. SPNP: stochastic Petri net package. In *Proceedings of the Third International Workshop on Petri Nets and Performance Models (PNPM'89)*, pages 142–151, Kyoto, Japan, 1989. IEEE Computer Society Press.

[5] Iain S. Duff, Albert M. Erisman, and John K. Reid. *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford, 1986.

[6] Laura C. Dutto, Wagdi G. Habashi, and Michel Fortin. Parallelizable block diagonal preconditioners for the compressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 117:15–47, 1994.

[7] Alan George and Joseph W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[8] Richard L. Klevans and William J. Stewart. From queuing networks to Markov chains: The XMARCA interface. *Performance Evaluation*, 24:23–45, 1995.

[9] Ivo Marek and Daniel B. Szyld. Local convergence of the (exact and inexact) iterative aggregation method for linear systems and Markov operators. *Numerische Mathematik*, 69:61–82, 1994.

[10] James O'Neil and Daniel B. Szyld. A block ordering method for sparse matrices. *SIAM Journal on Scientific and Statistical Computing*, 11:811–823, 1990.

[11] Jorge R. Paloschi, 1995. Centre for Process Systems Engineering, Imperial College, London, private communication.

[12] Bernard Philippe, Yousef Saad, and William J. Stewart. Numerical methods in Markov chain modeling. *Operations Research*, 40:1156–1179, 1992.

[13] Yousef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[14] Paul J. Schweitzer. A survey of aggregation-disaggregation in large Markov chains. In William J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 63–88, New York - Basel - Hong Kong, 1991. Marcel Dekker.

[15] M. E. Sezer and D. D. Šiljak. Nested epsilon decompostioins fo linear systems: Weakly coupled and overlapping blocks. *SIAM Journal on Matrix Analysis and Applications*, 12:521–533, 1991.

[16] William J. Stewart. MARCA: Markov chain analyzer. In William J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 37–61, New York - Basel - Hong Kong, 1991. Marcel Dekker.

[17] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, Princeton, New Jersey, 1994.

[18] William J. Stewart and Wei Wu. Numerical experiments with iteration and aggregation for Markov chains. *ORSA Journal on Computing*, 4:336–350, 1992.

[19] Daniel B. Szyld. Local convergence of (exact and inexact) iterative aggregation. In Carl D. Meyer and Robert J. Plemmons, editors, *Linear Algebra, Markov Chains and Queuing Models,* IMA Volumes in Mathematics and its Applications, Vol. 48, pages 137–143, New York – Berlin, 1993. Springer.

[20] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[21] Richard S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

[22] Geng Yang, Laura C. Dutto, and Michel Fortin. A parallelizable block Broyden method for nonlinear systems of equations. Technical Report R95-51, CEntre de Recherche en Calcul Appliqué (CERCA), Université de Montréal, April 1995.

Room TBA	***Workshop Chair***	
7:30 p.m.	***Mike Heroux***	Sparse and Parallel BLAS

Friday Evening's Workshop

Sparse and Parallel BLAS

Organizer: Michael Heroux

Abstract

In this workshop we will discuss the latest developments and efforts in Basic Linear Algebra Subprograms (BLAS) for sparse matrices and parallel computers. To begin the workshop, speakers will present their work in this area. Following this will be an open discussion on the presentations and related topics. Anyone wanting to speak may sign up during the conference before the time of the workshop.

Speakers

- Iain Duff, Rutherford Appleton Laboratory
  F90 Sparse BLAS
- Michael A. Heroux, Cray Research, Inc.
  Sparse BLAS Toolkit
- Roldan Pozo, Computing and Applied Mathematics Laboratory, NIST
  Object Oriented Sparse BLAS
- Tony Skjellum, Mississippi State University
  TBA

SATURDAY, APRIL 12TH

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*	*Room A*
8:00 - 8:30	R. Alchalabi	Multigrid Method Applied to the Solution of an Elliptic, Generalized Eigenvalue Problem
8:30 - 9:00	J. Dendy	Some Multigrid Algorithms for SIMD Machines
9:00 - 9:30	C. Douglas	Multigrid on Unstructured Grids Using an Auxiliary Set of Structured Grids
9:30 - 10:00	M. Griebel	Multiscale Iterative Methods, Coarse Level Operator Construction and Discrete Homogenization Techniques

Multigrid Method Applied to the Solution of an Elliptic, Generalized Eigenvalue Problem

Rifat M. Alchalabi * and Paul J. Turinsky**

*Advanced Modeling and Control Systems Department
BOC Group
575 Murray Hill, NJ 07974

**Department of Nuclear Engineering
P.O. Box. 7909
North Carolina State University
Raleigh, NC 27695-7909

Abstract

The work presented in this paper is concerned with the development of an efficient MG algorithm for the solution of an elliptic, generalized eigenvalue problem. The application is specifically applied to the multigroup neutron diffusion equation which is discretized by utilizing the Nodal Expansion Method (NEM). The underlying relaxation method is the Power Method, also known as the (Outer-Inner Method). The inner iterations are completed using Multi-color Line SOR, and the outer iterations are accelerated using Chebyshev Semi-iterative Method. Furthermore, the MG algorithm utilizes the consistent homogenization concept to construct the restriction operator, and a form function as a prolongation operator. The MG algorithm was integrated into the reactor neutronic analysis code **NESTLE**, and numerical results were obtained from solving production type benchmark problems.

Problem Description

In this work we consider the solution of the 3-dimensional, multigroup, eigenvalue neutron diffusion equation. The general form of this equation can be written as an elliptic partial differential equation as follows

$$-\bar{\nabla}\cdot\mathbf{D}_g\bar{\nabla}\Phi_g+\Sigma_{tg}\Phi_g=\sum_{g'=1}^{G}\Sigma_{sgg'}\Phi_{g'}+\frac{\chi_g}{k}\left(\sum_{g'=1}^{G}\nu_{g'}\Sigma_{fg'}\Phi_{g'}\right)$$

$$\text{for } g=1,2,\ldots G \qquad (1)$$

where the dependence of each quantity on the spatial coordinate **r** has been suppressed,. All the nuclear property related coefficient parameters,(i.e. D_g , Σ_g's , χ_g and ν_g) have non-negative values and are spatially piece-wise constant; and, k, Φ_g and g denote the multiplication factor (eigenvalue), flux (eigenfunction) and energy group, respectively. For 3-dimensional Cartesian geometry, Neumann boundary conditions are applied on two surfaces (x=0, y=0), and Dirichlet boundary conditions are applied on the other four surfaces ($x=x_{max}$, $y=y_{max}$, z=0, $z=z_{max}$).

Numerical Solution Method

Eqn.(1) is discretized by utilizing the Nodal Expansion Method (NEM) [1,2]. NEM can be thought of as a refinement to the finite-difference method (FDM), utilizing an improved approximation to the Laplacian operator. By applying NEM to Eqn.(1) we obtain the discretized matrix equation.

$$\mathbf{A}\,\Phi = \frac{1}{k}\,\chi\,\mathbf{F}\Phi \tag{2}$$

where energy group and spatial node dependence have been mapped to a single index. As normal, the matrix **A** has seven point spatial coupling; however, it is weakly nonsymmetric due to the NEM approximation of the Laplacian operator. Our objective is to solve Eqn.(2) for the dominant eigenvalue pair. This is accomplished using an Outer-Inner nested iterative strategy [3]. The Power Method is employed for the outer iteration, denoted for outer iteration count 'l' by

$$\mathbf{Q}\,\Psi^{(l)} = k^{(l)}\,\Psi^{(l)} \tag{3}$$

where

$$\Psi \equiv \mathbf{F}\,\Phi$$

$$\mathbf{Q} \equiv \mathbf{F}\,\mathbf{A}^{-1}\,\chi$$

and k is updated using the Raleigh Quotient. Iterative matrix **Q**, which is positive definite, has a significant role in determining the convergence rate of the power iterations [3]. Furthermore, the outer iteration is accelerated with the Chebyshev Semi-iterative method [2].

Eqn.(3), a fixed source matrix problem is solved iteratively, these iterations denoting the inner iteration process. The Multi-color Line SOR Method is employed with optimum relaxation parameter determined a priori via Gauss-Seidel iterations. Clearly some liberty has been taken in the selection of the iterative methods, since matrix **A** is not strictly symmetric.

Returning to the NEM discretization approximation, an improved Laplacian operator approximation is obtained as follows. Integrate Eqn.(1) in the two directions (e.g. (y,z)) transverse to a specific direction (e.g. x) over the span of a node. Now assume that the resulting terms involving Laplacian operator in the two transverse directions are known. This results in a one-dimensional (e.g. x) equation. NEM solves this equation for the transverse integrated flux (e.g. node-wise (y,z) flux average as a function of x) using polynomial trial functions. Operating upon this solution by the Laplacian operator, which requires no further approximation, provides an accurate estimate of this operation which is employed to correct the spatial coupling coefficients that originate from the FDM approximation used to obtain Eqn.(2). This can be implemented in a nested iterative manner (i.e. NEM-(Outer-Inner) iterations), which is a nonlinear iterative technique since the Outer-Inner iterations utilize matrix **A**, which is iteratively updated by the NEM iterations.

Multigrid Implementation

Several algorithms incorporating MG have been proposed for the solution of the few-group neutron diffusion equation. Alcouffe [4] implemented a MG method that is imbedded inside the outer iteration. Finnemann, at al. [5] implemented several multi-level techniques for the solution of the nodal diffusion equation. The most sophisticated method he employed was to imbed the NEM-(Outer-Inner) nested iterations inside the MG cycle; however, this method lacked the robustness and suffered from solution divergence for realistic benchmarks. Zaslavsky [6] proposed an adaptive algebraic MG for reactor criticality calculations. Similar to Alcouffe's method, the MG algorithm were imbedded inside the outer iteration. However, the outer iteration was solved via the inverse iteration method, and the interpolation operator was a variant of the form function.

In our MG implementation, it is used to accelerate the Power Method (i.e. the outer iteration) since iterative matrix **Q**'s dominance ratio is close to 1.0. Thus NEM iterations are performed only on the finest grid level. The MG schedule controls the onset of each MG (i.e. what NEM iterations to complete MG at). Note that the coefficient matrix within any given MG cycle is constant since no NEM correction updates are applied. The advantage of this approach is the high computational efficiency, and the relative simplicity in both implementation and analysis.

Due to the inherent recursion of the employed MG algorithm, the two-grid algorithm will be sufficient to describe the algorithm. The MG algorithm proceeds first at the fine-grid level G^M for a few NEM iterations until a specified tolerance on a relative L_2 norm is satisfied. Thus an approximate solution, $\tilde{\Phi}^{M(n)}$, is obtained at NEM iteration count (n). This step is followed by descending to the next coarser-grid level G^{M-1}. The coarse-grid operators are constructed in such a way that they preserve the current, approximate fine-grid solution $\tilde{\Phi}^{M(n)}$ on the coarse-grid level G^{M-1} (i.e. consistent homogenization). The initial solution estimate on the coarse-grid level, G^{M-1} is obtained by simply volume weighting the approximate fine-grid solution, $\tilde{\Phi}^{M(n)}$, denoted as

$$\Phi^{M-1(n,l)} = I_M^{M-1} \tilde{\Phi}^{M(n)} \qquad \text{for } l=0 \tag{4}$$

This is consistent with using a nodal method, where volume average versus point values enter the formulation. At this coarse-grid level several outer iterations are performed to satisfy a specified error tolerance on an L_2 relative norm. The approximate solution to the above coarse-grid problem, $\tilde{\Phi}^{M-1(n)}$ is used along with the approximate fine-gird solution during ascending to construct the updated fine-grid solution estimate for fine-grid level G^M as follows

$$\Phi^{M(n,0)} = I_{M-1}^{M(n)} \tilde{\Phi}^{M-1(n)} \tag{5}$$

where the prolongation operator, $I_{M-1}^{M(n)}$, is defined based on the "form function" concept, borrowed from the pin power reconstruction methodology commonly used in reactor physics computations. Mathematically the prolongation operator can be expressed as

$$\left(\mathbf{I}_{M-1}^{M^{(n)}}\right)_{i'i} = \frac{\left(\tilde{\Phi}^{M^{(n)}}\right)_{i'}}{\left(\mathbf{I}_{M}^{M-1}\tilde{\Phi}^{M^{(n)}}\right)_{i}} \tag{6}$$

for " i` " denoting a fine-node contained within a coarse-node " i ", and zero otherwise. Substituting Eqns.(4) and (6) into Eqn.(5), and performing mathematical manipulation, the following is obtained:

$$\Phi^{M^{(n,0)}} = \tilde{\Phi}^{M^{(n)}} + \mathbf{I}_{M-1}^{M^{(n)}}\left(\tilde{\Phi}^{M-1^{(n)}} - \Phi^{M-1^{(n,0)}}\right) \tag{7}$$

It is clear from Eqn.(7) that the prolongation operator operates on the solution correction rather than the past solution. Thus the newly introduced interpolation errors are limited to the solution correction only. The advantage in using the "form function" is that it assures the non-smooth components of the error it introduces can be attenuated rapidly with a few relaxation sweeps, since these factors represent a local (fine-grid) correction. Thus these factors do not amplify the smooth error components. The choice of the prolongation operator, in conjunction with the formerly described restrictive operator, makes the current algorithm behave in a similar fashion to the efficient FAS MG method.

As noted above, the coarse-grid operator is generated employing consistent homogenization. Mathematically this can be stated as

$$\mathbf{B}^{M-1^{(n)}}\,\Phi^{M-1^{(n,0)}} = \mathbf{B}^{M-1^{(n)}}\,\mathbf{I}_{M}^{M-1}\,\tilde{\Phi}^{M^{(n)}} = \mathbf{I}_{M}^{M-1}\,\mathbf{B}^{M^{(n)}}\,\tilde{\Phi}^{M^{(n)}} \tag{8}$$

where **B** denotes any of the matrices in Eqn.(2). For multiplier operators of Eqn.(1) (e.g. $\Sigma_{tg}\,\Phi_g$), that produce diagonal components of **B**,

$$\left(\mathbf{B}^{M-1^{(n)}}\right)_{j,j'} = \frac{\left(\mathbf{I}_{M}^{M-1}\,\mathbf{B}^{M^{(n)}}\,\tilde{\Phi}^{M^{(n)}}\right)_{j,j'}}{\left(\mathbf{I}_{M}^{M-1}\,\tilde{\Phi}^{M^{(n)}}\right)_{j,j'}} \tag{9}$$

which physically equates to preserving coarse-grid neutron interaction rates as determined by fine-grid results. For the Laplacian operator of Eqn.(1), that produces the off-diagonal components of **B** (specifically **A**), these spatial coupling coefficients are determined in the same manner as for the NEM. Physically, this corresponds to preserving neutron leakage out of the coarse-grid node as determined by fine-grid results. Clearly the eigenvalue is preserved on the coarse-grid. The advantage of using a consistent homogenization is that no residual due to coarse-grid operator error is introduced, retaining the original eigenvalue problem structure.

During the course of descending in the MG Cycle (coarsening the grids), the matrices are saved for each level so they may be reused when that level is revisited during ascending (refining the grid).

Numerical Results and Discussion

In this section several numerical results are presented and analyzed. The emphasis of this section is to present a comparison of the computational efficiency for different iterative solution strategies. The test problem presented corresponds to a typical production type problem associated with analyzing a light water power reactor core. Due to the complexity of this problem, only the basic problem attributes are described in Table 1, with the full problem description available in Ref. [3]. The MG algorithm employed utilizes a fixed scheduling. The convergence criteria used to control the number of Outer iterations (number of relaxation sweeps) at any grid level is

$$\frac{\left\| \Psi^{M^{(n+1,0)}} - \tilde{\Psi}^{M^{(n)}} \right\|_2}{\left\langle \Psi^{M^{(n+1,0)}}, \tilde{\Psi}^{M^{(n)}} \right\rangle^{1/2}} \leq \varepsilon_{\Psi}^{s} \qquad \text{for s at M} \tag{10}$$

and

$$\frac{\left\| \Psi^{M\text{-}1^{(n,l+1)}} - \Psi^{M\text{-}1^{(n,l)}} \right\|_2}{\left\langle \Psi^{M\text{-}1^{(n,l+1)}}, \Psi^{M\text{-}1^{(n,l)}} \right\rangle^{1/2}} \leq \varepsilon_{\Psi}^{s} \qquad \text{for s at M-1} \tag{11}$$

where ε_{Ψ}^{s} is the L_2 norm stage termination criteria for stage number "s" in the MG schedule.

Figure 1 presents the true L2 error norm versus CPU time for four different iterative cases. Based upon tightly converged numerical solutions, denoted $\Phi_g^{M'}$, the true L_2 error norm of grid level $G^{M'}$ is given by

$$L_2^{M'} = \frac{\left\| \Phi_g^{M'^{(n,l)}} - \Phi_g^{M'} \right\|_2}{\left\| \Phi_g^{M'} \right\|_2} \qquad \text{for } M' = M \text{ or } M\text{-}1 \tag{12}$$

Note when interpreting Figure 1, the grid level M` changes with the CPU time. The MG method utilizes a two-level grid, four V-cycles to accelerate the solution. Since the function of the Outer-Inner iteration inside the MG cycle is to dampen the high frequency error components rather than to solve the problem accurately, the number of inner iterations inside each outer iteration is restricted to two iterations [7]. Furthermore, when the Chebyshev method is used to accelerate the outer iterations, the iterative matrix dominance ratio estimate is reduced by 2% to accelerate high frequency error damping and to insure that the estimate stays below the true dominance ratio estimate. The termination criteria for the different MG stages are illustrated in Figure 2. Table 2 presents a summary of the result for the four basic iterative cases. The total number of iterations reported for cases with MG on are for the sum of the fine and coarse mesh iterations. Table 2 shows that the total code CPU time is reduced by a factor of 2.15 and 1.60, respectively, for the case with Chebyshev acceleration off and on. Note the computational platform used is a Sun SPARCstation 20, equipped with HyperSparc processor. Numerical experiments revealed that the optimum number of NEM iterations is 8. Since the MG cycles operate inside the NEM iteration, it follows that the total NEM CPU time should not be effected by the MG acceleration. Thus a better measure for the MG computational efficiency is the reduction in CPU time for the Outer-Inner iterative portion of the calculation. Table 2 shows a CPU time reduction for Outer-

Inner iterations of 3.52 and 2.70, respectively, for the cases with Chebyshev acceleration off and on. To evaluate the numerical effectiveness of both the usage of consistent homogenization for the Laplacian operator as a restriction operator and the form factor as a prolongation operator, these two operators were disabled separately, and the L_2 norm of the exact error versus CPU time for both cases, with Chebyshev acceleration on, are plotted in Figure 3. Note in Figure 3, disabling the restriction operator is equivalent to disabling the improved approximation in the Laplacian operator, hence not preserving the solution on the coarse-grid problem. This case is labeled in Figure 3, " Without Full Homogenization".

References

[1] K. Smith, "Nodal Method Storage Reduction by Non-Linear Iteration," *Trans. Am. Nucl. Soc.*, 44, 265 (1983).

[2] P. Turinsky, R. Alchalabi, P. Engrand, H. Sarsour, F. X. Faure, and W. Guo "*NESTLE: A Few Group Neutron Diffusion Equation Solver Utilizing the Nodal Expansion Method for Eigen-Value, Adjoint, Fixed-Source Steady-State and Transient Problems,*" U.S. Department of Energy, Idaho Operations Office, EGG-NRE-11406, June (1994).

[3] R. Alchalabi, "Development of Neutronic Core Physics Simulator on Advanced Computer Architecture," Master of Science Thesis, N.C. State University (1991).

[4] R. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," *Advances in Reactor Computations, Topical Meeting Proceedings,* Salt Lake City, UT, 340, March (1983).

[5] H. Finnemann, R. Boer, R. Muller, and Y.Kim, "Multi-level Techniques for the Acceleration of Nodal Reactor Calculations," *Proceedings of the International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics,* Pittsburgh, PA, April 28-May 2 (1991).

[6] L. Zaslavsky, "An Adaptive Algebraic Multigrid for Reactor Criticality Calculation," *SIAM J. SCI Comput.* **16**, No. 4, 840, July (1995).

[7] Personal communication with Dr. Steven McCormick and Dr. John Ruge. (University of Colorado at Boulder), May 1994.

Description	Test Case
Fine- Grid	18 X 18 X 18 = 5832
Coarse-Grid	10 X 10 X 10 = 1000
Fine-Grid Outer Iterative Matrix Dominance Ratio	0.980

Table 1. Basic Problem Attributes

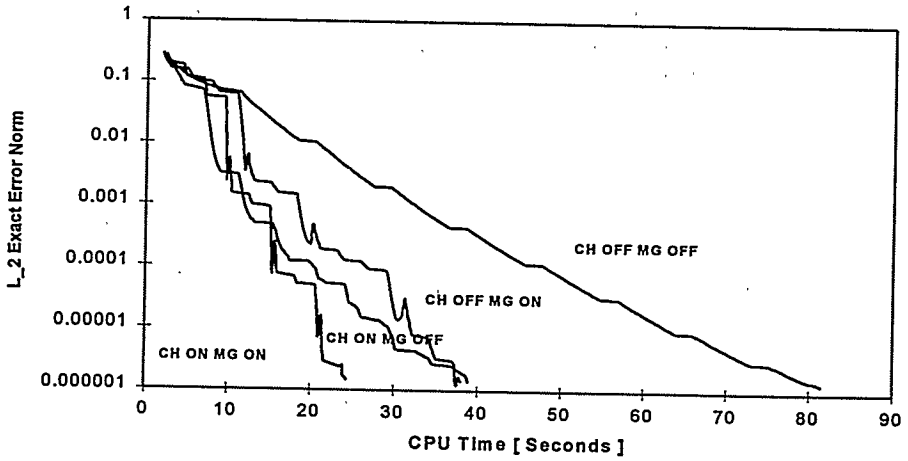


Figure 1 Computational Time for the Four Iterative Cases

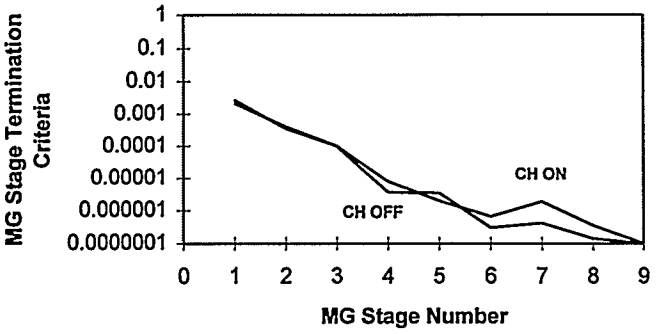


Figure 2 MG Stage Termination Criteria

Description	MG OFF Cheby OFF	MG OFF Cheby ON	MG ON Cheby OFF	MG ON Cheby ON
No. Of NEM Iterations	8	8	10	8
No. Of Outer Its	401	134	350	105
No. Of Inner Its.	3208	1072	1400	420
NEM CPU Time [Sec's]	14.25	14.37	17.77	14.20
Outer CPU Time [Sec's]	13.09	5.35	7.21	4.18
Inner CPU Time [Sec's]	52.15	17.16	11.32	4.17
Total Code CPU Time [Sec's]	81.40	38.78	37.92	24.17
Total Outer-Inner Iterations CPU Time [Sec's]	65.24	22.51	18.53	8.35

Table 2. Numerical Results for the Four Iterative Cases

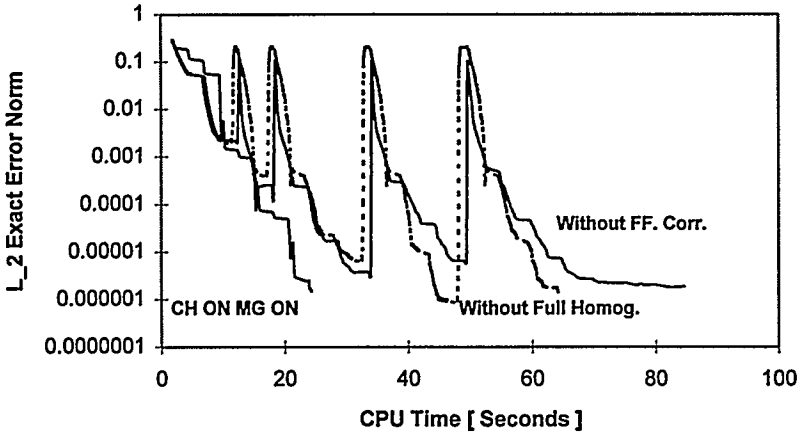


Figure 3 Computational Time Comparison for the Reference, Without Full Homogenization, and Without Form Factor Correction Cases

SOME MULTIGRID ALGORITHMS FOR SIMD MACHINES

J. E. Dendy, Jr.
Thoretical Division
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Abstract. Previously a semicoarsening multigrid algorithm suitable for use on SIMD architectures was investigated. Through the use of new software tools, the performance of this algorithm has been considerably improved. The method has also been extended to three space dimensions. The method performs well for strongly anisotropic problems and for problems with coefficients jumping by orders of magnitude across internal interfaces. The parallel efficiency of this method is analyzed, and its actual performance on the CM-5 is compared with its performance on the CRAY-YMP. A standard coarsening multigrid algorithm is also considered, and we compare its performance on these two platforms as well.

Multigrid on Unstructured Grids Using an Auxiliary Set of Structured Grids

Craig C. Douglas*
Sachit Malhotra
Martin H. Schultz

Yale University
Department of Computer Science
P.O. Box 208285
New Haven, CT 06520-8285
USA

Unstructured grids do not have a convenient and natural multigrid framework for actually computing and maintaining a high floating point rate on standard computers. In fact, just the coarsening process is expensive for many applications. Since unstrucutured grids play a vital role in many scientific computing applications, many modifications have been proposed to solve this problem. One suggested solution is to map the original unstructured grid onto a structured grid. This can be used as a fine grid in a standard multigrid algorithm to precondition the original problem on the unstructured grid. We show that unless extreme care is taken, this mapping can lead to a system with a high condition number which eliminates the usefulness of the multigrid method.

Theorems with lower and upper bounds are provided. Simple examples show that the upper bounds are sharp.

MULTISCALE ITERATIVE METHODS, COARSE LEVEL OPERATOR CONSTRUCTION AND DISCRETE HOMOGENIZATION TECHNIQUES

MICHAEL GRIEBEL
INSTITUT FÜR INFORMATIK, TECHNISCHE UNIVERSITÄT MÜNCHEN,
ARCISSTRASSE 21, D-80290 MÜNCHEN, GERMANY
E-MAIL: GRIEBEL@INFORMATIK.TU-MUENCHEN.DE

For problems which model locally strong varying phenomena on a micro-scale level, the grid for numerical simulation can not be chosen sufficiently fine enough due to reasons of storage requirements and numerical complexity. A typical example for such kind of a problem is the diffusion equation with strongly varying diffusion coefficients as it arises as Darcy law in reservoir simulation and related problems for flow in porous media.

Therefore, on the macro-scale level, it is necessary to work with averaged equations which describe directly the large-scale behavior of the problem under consideration. In the numerical simulation of reservoir performance this is achieved e.g. by renormalization or homogenization, as simpler approaches like the arithmetic, geometric or harmonic mean turn out to be invalid for systems with strong permeability variations.

We apply the Galerkin approximation, used in multi grid methods for determining coarse grid equations, as a discrete method for calculating such averaged equations. The discretization of an elliptic differential equation can be interpreted as a certain kind of averaging or filtering, where smaller scales than the respective grid size are eliminated. Analogously, the Galerkin coarse grid equations resemble discrete averaged equations, modelling the influence of the scales smaller than the grid size of the actual level. Methods known from multi grid context as operator- or matrix-dependent prolongations and Schur complement approximations lead to energy-dependent averaging procedures and to averaged equations which describe the large-scale behavior of the problem in discrete form.

We explain our coarse level operator construction (which involves some sort of incomplete factorization approach) and our discrete homogenization technique, discuss its properties and compare it with the established averaging, renormalization and homogenization methods. Additionally we consider the convergence behaviour of the corresponding multilevel iterative methods. We give results from numerical experiments for the diffusion equation with various types of diffusion fields.

Topic: *Applications*	*Session Chair:* *TBA*	*Room B*
8:00 - 8:30	H.C. Chen	Embedding SAS Approach Into Conjugate Gradient Algorithms for Asymmetric 3D Elasticity Problems
8:30 - 9:00	M. Clemens	Iterative Methods for the Solution of Very Large Complex-Symmetric Linear Systems of Equations in Electrodynamics
9:00 - 9:30	T. Cwik	Matrix Equation Decomposition and Parallel Solution of Systems Resulting from Unstructured Finite Element Problems in Electromagnetics
9:30 - 10:00	S.W. Bova	Iterative Solution of the Semiconductor Device Equations

EMBEDDING SAS APPROACH INTO CONJUGATE GRADIENT ALGORITHMS FOR ASYMMETRIC 3D ELASTICITY PROBLEMS *

HSIN-CHU CHEN †, AHMED SAMEH ‡ AND NAZIR A. WARSI†

ABSTRACT

In this paper, we present two strategies to embed the SAS (symmetric-and-antisymmetric) scheme into conjugate gradient (CG) algorithms to make solving 3D elasticity problems, with or without global reflexive symmetry, more efficient. The SAS approach is physically a domain decomposition scheme that takes advantage of reflexive symmetry of discretized physical problems, and algebraically a matrix transformation method that exploits special reflexivity properties of the matrix resulting from discretization. In addition to offering large-grain parallelism, which is valuable in a multiprocessing environment, the SAS scheme also has the potential for reducing arithmetic operations in the numerical solution of a reasonably wide class of scientific and engineering problems. This approach can be applied directly to problems that have global reflexive symmetry, yielding smaller and independent subproblems to solve, or indirectly to problems with partial symmetry, resulting in loosely coupled subproblems. The decomposition is achieved by separating the reflexive subspace from the antireflexive one, possessed by a special class of matrices A, $A \in \mathcal{C}^{n\times n}$, that satisfy the relation $A = PAP$ where P is a reflection matrix (symmetric signed permutation matrix).

Although there are a great number of problems with global reflexive symmetry, many other problems are asymmetric. For such asymmetric problems, direct application of this approach to the whole problem to yield entirely independent subproblems does not seem possible. Therefore, we resort to indirect applications so that the advantage of this approach can be carried over. This is well situated when it is employed in conjunction with iterative schemes such as the (preconditioned) CG algorithms. There are two different approaches to achieving this goal. One is to construct a preconditioner for the CG algorithm in such a way that the preconditioner, as close to the original matrix as possible, has the desired reflexivity property and then apply SAS to the preconditioning linear system. The other is to split the matrix into as many matrices as desired so that most of them possess some form of reflexivity property, and then apply SAS only to those with the reflexivity property

* This work was partially supported by the U.S. Department of Energy under Grant No. DOE-DE-FG02-85ER25001.

† Army Center of Excellence in Information Sciences, Department of Computer and Information Sciences, Clark Atlanta University, 223 James P. Brawley Dr. at Fair St. SW, Atlanta, GA 30314.

‡ Department of Computer Science, University of Minnesota at Minneapolis, 4-192 EE/Sci. Bldg., 200 Union St. SE, Minneapolis, MN 55455.

in performing matrix-vector multiplications, with or without a preconditioner. The first approach is very useful when the physical problem to be solved is only slightly perturbed from symmetry since, in such a case, a good preconditioner can easily be constructed. The advantage of the second approach lies in its flexibility, in the sense that it can be adapted to problems with very complex domains and properties due to the freedom it offers to split the matrix. It is especially suitable for the element-by-element computations, often employed in the CG algorithm for finite element analysis, because it has the ability to exploit local symmetry either at the subdomain level or at the element level. The application of the second approach to a 3D elasticity problem using (preconditioned) CG algorithms will be presented to demonstrate its effectiveness.

1. SAS Approach. The main idea of the SAS approach is the exploitation of the special properties of reflexive matrices, which arise naturally from discretized physical problems with reflexive symmetry [ChSa89a]. This is a domain reduction approach using symmetry [DoSm89, DoMa92] and a special application of group-theoretic methods [FaSt92]. In this paper, we extend this idea to handle problems without physical symmetry by embedding SAS into the CG/PCG algorithm.

Before presenting our extension, we first state three basic lemmas involved in the SAS approach as shown below, where the matrix P is assumed to be some *nontrivial* ($P \neq \pm I$) reflection matrix of dimension n.

- Lemma 1: Given a nonsingular linear system $Ax = f$, $A \in \mathcal{C}^{n\times n}$ and $f \in \mathcal{C}^n$, if A is reflexive with respect to P, then $x = Px$ (or $x = -Px$) if and only if $f = Pf$ (or $f = -Pf$).
- Lemma 2: Any vector $b \in \mathcal{C}^n$ can be decomposed into two parts, u and v, $u + v = b$, such that $u = Pu$ (reflexive) and $v = -Pv$ (antireflexive).
- Lemma 3: Any matrix $A \in \mathcal{C}^{n\times n}$ can be decomposed into two parts, U and V, $U + V = A$, such that $U = PUP$ and $V = -PVP$.

There are three steps involved in this approach. The first step is to decompose f into its reflexive part u and antireflexive part v (Lemma 2). This can be done by taking $u = (f+Pf)/2$ and $v = (f-Pf)/2$. We then reduce $Ay = u$ and $Az = v$ to smaller systems by dropping redundant unknowns in y and z (Lemma 1) from them and solve the reduced systems, instead of solving $Ax = f$ directly. In the last step, we retrieve the solution x from y and z. This decomposition in essence leads to a transformation that block-diagonalizes the matrix A. Note that P is symmetric and has only two distinct eigenvalues. Therefore, corresponding to each P there exists an orthogonal transformation matrix Q that can block-diagonalize A [HoJo85, pp.50-52] (via the transformation Q^TAQ) into two independent submatrices since A and P commute. The explicit form of Q certainly depends on P. One of the frequently

encountered forms of the matrix pair for P and Q is

$$P = \begin{bmatrix} 0 & S & 0 \\ S & 0 & 0 \\ 0 & 0 & I_2 \end{bmatrix} \quad \text{and} \quad Q = \frac{1}{\sqrt{2}} \begin{bmatrix} I_1 & S & 0 \\ -S & I_1 & 0 \\ 0 & 0 & \sqrt{2}I_2 \end{bmatrix}$$

where S is a diagonal matrix whose diagonal elements are either 1 or -1, I_1 and I_2 are identity matrices.

To employ this approach for efficient computations, however, we need to know the reflection matrix P beforehand. Although it is not trivial in most cases to see directly from the matrix entries whether a matrix is reflexive or not, the reflexivity of a matrix and its associated reflection matrix P (given it is reflexive) can usually be determined from the original physical problem and its discretization; see [ChSa89a] for examples. Once P is identified, the decoupling of the original system $Ax = f$ to two smaller independent subsystems follows immediately from the orthogonal transformation of A to $\tilde{A} = Q^T A Q$ or from using Lemmas 1 and 2 directly.

2. SAS-embedded CG/PCG Algorithm. To take advantage of SAS for solving 3D elasticity problems without reflexive symmetry, we proposed to embed SAS into the CG/PCG algorithm for solving the preconditioning system and/or for performing the matrix-vector multiplication using either the element-by-element or the subdomain-by-subdomain approach. Let $Kx = f$, $K \in \mathcal{R}^{n \times n}$ and $f \in \mathcal{R}^n$, be the algebraic linear system resulting from the finite element discretization, where K is assumed to be symmetric positive definite (SPD). Other than being SPD, the matrix K need not have any reflexivity property. Shown below is a PCG algorithm [JoMP83] where A is a nonsingular preconditioner for K, x_0 is the initial approximation to x, and the symbol $(u,\ v)$ denotes the inner product of vectors u and v.

PCG Algorithm:

- Compute the residual $r_0 = b - Kx_0$.
- Solve $Ad_0 = r_0$ for d_0 and set $p_0 = d_0$.
- For $i = 0$, 1, ..., k-1, do the following

  $\alpha_i = (r_i,\ d_i)/(p_i,\ Kp_i)$

  $x_{i+1} = x_i + \alpha_i p_i$

  $r_{i+1} = r_i + \alpha_i K p_i$

  Solve $Ad_{i+1} = r_{i+1}$ for d_{i+1}

  $\beta_i = (r_{i+1},\ d_{i+1})/(r_i,\ d_i)$

  $p_{i+1} = d_{i+1} + \beta_i p_i$

  until convergence is achieved.

The classical conjugate gradient (CG) algorithm [HeSt52] without preconditioning corresponds to the case when $A = I$. One of the main advantages of using CG over direct solvers to solve linear systems resulting from finite element discretizations is that, in addition to its simplicity, CG allows for its matrix-vector multiplications $q_i = Kp_i$ to be performed on an element-by-element basis, thus eliminating the need to assemble the global stiffness matrix K. When PCG is employed, the main computational issue involved in the algorithm is the effectiveness of the preconditioner A. Without a good preconditioner, PCG may not converge fast enough to overcome the extra computational overhead required in solving the preconditioning systems. Therefore, a good candidate for an effective preconditioner should be one that not only can reduce the number of iterations but also allows the preconditioning linear system $Ad_i = r_i$ to be solved as efficiently as possible so that the total solution time can be substantially reduced. In addition, when the algorithm is to be implemented on a multiprocessor, the matrix A should also be chosen in such a way that the preconditioning linear system can be solved in parallel. Such a good preconditioner is difficult to derive without knowledge of the physical problem and/or its associated matrix.

As mentioned earlier, there are two approaches to embedding SAS into the PCG algorithm. In the first approach, we choose the preconditioner A to be reflexive on one hand (using Lemma 3 for example) and to be as close to the original matrix K as possible on the other. This of course involves a proper choice of P. Physically, this corresponds to using a slightly perturbed problem with reflexive symmetry to precondition the original asymmetric problem. Let Q be the transformation matrix associated with P. Then, instead of solving the preconditioning system $Ad_i = r_i$, $i = 0, 1, 2, \cdots$, we solve the SAS-transformed system $\tilde{A}\tilde{d}_i = \tilde{r}_i$ where

$$\tilde{A} = Q^T A Q, \ \tilde{d}_i = Q^T d_i, \ \tilde{r}_i = Q^T r_i.$$

The resulting algorithm can be obtained simply by replacing the step of solving $Ad_i = r_i$ with the following three steps:

1. Decomposing r_i into $\tilde{r}_i$: $\tilde{r}_i = Q^T r_i$.
2. Solving $\tilde{A}\tilde{d}_i = \tilde{r}_i$ for $\tilde{d}_i$.
3. Retrieving d_i from $\tilde{d}_i$: $d_i = Q\tilde{d}_i$.

Note that the decomposition of A into $\tilde{A}$ is performed once and for all, regardless of the number of iterations. The decomposition of r_i into $\tilde{r}_i$ and the retrieval of d_i from $\tilde{d}_i$, however, need be performed in each iteration. This results in an overhead in each iteration. Fortunately, this overhead is usually negligible due to the simplicity and sparsity of the transformation matrix Q. The effectiveness of this approach, thus, depends mainly on two factors. The first is the computational savings that can be gained from solving the decomposed subsystems

instead of from the undecomposed preconditioning system. The second is the closeness of the preconditioner to the original problem. In practice, this approach can be very useful when the problem to be solved is only slightly perturbed from some other problem that allows for a direct application of the SAS approach [ChSa89b].

When such a preconditioner cannot be found, we resort to our second approach by embedding SAS into the algorithm for performing the matrix-vector multiplication involved in each iteration. This can be done either at the subdomain or at the elment level. Algebraically, we split the original matrix K into, say, $s+1$ matrices: K_0, K_1, $\cdots$, K_s such that K_i is reflexive with respect to some nontrivial reflection matrices P_i for $i = 1,\ 2,\ \cdots,\ s$. In other words

$$K = K_0 + \sum_{i=1}^{s} K_i \quad \text{with} \quad K_i = P_i K_i P_i, \quad i \neq 0.$$

The matrix K_0, which may or may not have any special property, can be considered as the remainder of the splitting since K_i for $1 \leq i \leq s$ are all reflexive.

As previously explained, corresponding to each P_i there exists a transformation matrix Q_i that can block-diagonalize K_i into one that consists of two diagonal blocks, say $\tilde{K}_i$, $\tilde{K}_i = Q_i^T K_i Q_i$. The matrix-vector multiplication Kp can, therefore, be expressed as

$$Kp = K_0 p + \sum_{i=1}^{s} K_i p = K_0 p + \sum_{i=1}^{s} Q_i \tilde{K}_i Q_i^T p.$$

In other words, we replace $q_i = K_i p$, $i \neq 0$, with the following three-step multiplication:

$$u_i = Q_i^T p, \quad v_i = \tilde{K}_i u_i, \quad q_i = Q_i v_i.$$

It should be mentioned that for this replacement to be effective, the three-step multiplication for $q_i = K_i p$ must be more efficient than its single-step multiplication, which is usually the case when the transformation can be accomplished by the SAS approach.

3. Applications to Asymmetric 3D Elasticity Problems. In this section, we present our application of SAS using the second approach discussed in the previous section to an asymmetric 3D orthotropic elasticity problem whose differential equations are described by

$$\mathcal{L}^T D \mathcal{L} \sigma + \mathbf{b} = \mathbf{0}$$

where σ, $\mathbf{b}$, $\mathcal{L}$, and D are the stress vector, body force vector, differential operator matrix, and the material property matrix, respectively:

$$\sigma = [\ \sigma_{xx},\ \sigma_{yy},\ \sigma_{zz},\ \sigma_{xy},\ \sigma_{xz},\ \sigma_{yz}\ ], \quad \mathbf{b} = [\ b_x,\ b_y,\ b_z\ ],$$

TABLE 1

CPU time in seconds on the Alliant FX/8

Approach	No. of processors			
	1	2	4	8
CG alone	510.8	306.1	160.8	110.1
CG + SAS	398.0	219.7	117.6	77.8
PCG alone	356.2	216.1	112.4	79.7
PCG + SAS	285.2	160.2	85.2	54.5

$$L_1 = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ \partial/\partial z & 0 & \partial/\partial x \\ 0 & \partial/\partial z & \partial/\partial y \end{bmatrix}, \text{ and } D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & 0 & 0 & 0 \\ d_{12} & d_{22} & d_{23} & 0 & 0 & 0 \\ d_{13} & d_{23} & d_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{66} \end{bmatrix}.$$

The physical problem used for our experiments is a cantilevel beam with additional support from two springs of different stiffness K_1 and K_2, as shown in Figure 1. A concentrated load P and a uniformly distributed bending moment M are applied to the beam at the right end. The beam is modeled as a 3D elasticity problem and discretized with a 15 (spacings) $\times$ 7 $\times$ 7 grid. The actual numerical values for the dimensions/parameters/constants related to this beam [ChSa89b] are not essential in this paper and, thus, omitted. The element type employed for our discretization is the eight-node rectangular hexahedral element which is shown in Figure 2. This type of elements have three planes of symmetry and their element stiffness matrices have been shown to possess a three-level reflexivity property [ChSa89a]. In our experiments, we group the elements into 15 subdomains as shown in Figure 1 and embed SAS into the CG/PCG algorithm at the subdomain level, using the fact that subdomains 1 through 14 all have three planes of symmetry and subdomain 15 has two planes of symmetry due to the boundary conditions at the fixed end. The two springs are considered as the 16^{th} subdomain, which does not have any symmetry. The performance of the CG/PCG algorithm with and without embedding the SAS approach is presented in Table 1, where the CG algorithm does not employ any preconditioner while the PCG algorithm uses the main diagonal as the preconditioner. As seen from this table, it is clear that the embedding of SAS into the algorithms makes them more efficient. This is due to the reduction in arithmetic operations induced by SAS in performing the matrix-vector multiplications of the CG/PCG algorithm.

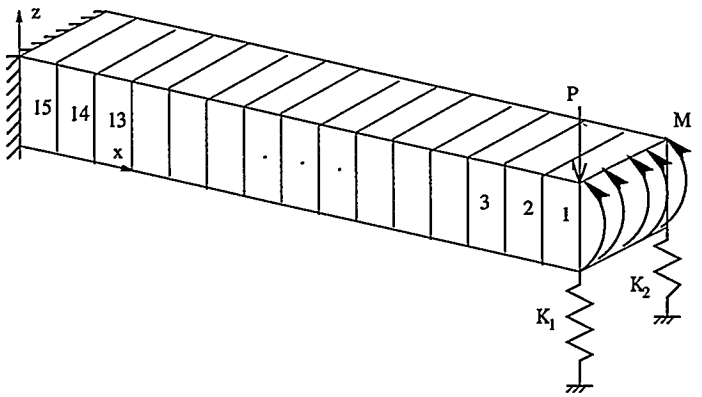


FIG. 1. *A beam with supporting springs.*

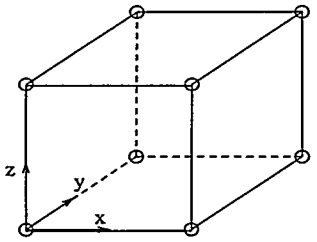


FIG. 2. *A rectangular hexahedral element.*

REFERENCES

[ChSa89a] H-C. Chen and A. Sameh, *A matrix decomposition method for orthotropic elasticity problems*, SIAM J. Matrix Anal. Appl., 10(1), 1989, pp. 39-64.

[ChSa89b] H-C. Chen and A. Sameh, *A domain decomposition method for 3D elasticity problems*, in Applications of Supercomputers in Engineering: Fluid Flow and Stress Analysis Applications, C.A. Brebbia and A. Peters, eds., Computational Mechanics Publications, Southampton University, Southampton, England, 1989, pp.171-188.

[DoMa92] C.C. Douglas and J. Mandel, *An abstract theory for the domain reduction method*, Computing 48, 1992, pp.73-96.

[DoSm89] C.C. Douglas and B.F. Smith, *Using symmetries and antisymmetries to analyze a parallel multigrid algorithm: the elliptic boundary value problem case*, SIAM J. Numer. Anal., 26(6), 1989, pp.1439-1461.

[FaSt92] A. Fassler and E. Stiefel, Group Theoretical Methods and Their Applications, Birkhauser, Boston-Basel-Berlin, 1992.

[HoJo85] R.A. Horn and C.A. Johnson, Matrix Analysis, Cambridge University Press, New York, 1985.

[HeSt52] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand. 49, pp.409-436, 1952.

[JoMP83] O.J. Johnson, C.A. Micchelli, and G. Paul, *Preconditioners for conjugate gradient calculations*, SIAM J. Numer. Anal., 20(2), 1983.

Iterative Methods for the Solution of Very Large Complex Symmetric Linear Systems of Equations in Electrodynamics

Markus Clemens
and
Thomas Weiland
Fachbereich 18 Elektrische Nachrichtentechnik, Fachgebiet Theorie Elektromagnetischer Felder,
Technische Hochschule Darmstadt,
Schloßgartenstr. 8, D-64289 Darmstadt, Germany

Abstract

In the field of computational electrodynamics the discretization of Maxwell's equations using the Finite Integration Theory (FIT) yields very large, sparse, complex symmetric linear systems of equations. For this class of complex non-Hermitian systems a number of conjugate gradient-type algorithms is considered. The complex version of the biconjugate gradient (BiCG) method by Jacobs can be extended to a whole class of methods for complex-symmetric algorithms SCBiCG(Γ, n), which only require one matrix vector multiplication per iteration step. In this class the well-known conjugate orthogonal conjugate gradient (COCG) method for complex-symmetric systems corresponds to the case $n = 0$. The case $n = 1$ yields the BiCGCR method which corresponds to the conjugate residual algorithm for the real-valued case. These methods in combination with a minimal residual smoothing process are applied separately to practical 3D electro-quasistatical and eddy-current problems in electrodynamics. The practical performance of the SCBiCG methods is compared with other methods such as QMR and TFQMR.

1. Introduction. In the field of computational electrodynamics the discretization of Maxwell's Equations using the Finite Integration Theory (FIT) yields very large, sparse linear systems of equations $Ax = b$. The rank of these systems ranges from 10^3 up to 10^6 unknows. The focus of interest in this paper is the efficient solution of a special class of non-Hermitian systems as they occur in time-harmonic eddy current problems including materials with finite electric conductivity and in the so-called electro-quasistatic approach where a complex scalar potential equation has to be solved. The linear systems resulting from the discretization process have in common that they are usually very large, sparse and complex symmetric. For such sparse non-Hermitian linear systems efficient iteration algorithms have been developed over the past years. Some of them, e.g. the GMRES method [12], have very desirable global residual minimization properties, but these are usually connected with the requirement to store a vast number of subspace basis vectors. Since the truncated versions of these algorithms usually have much less competitive convergence properties, the class of Krylov subspace methods based on the Lanczos biorthogonalization process becomes attractive due to its short vector recurrences. For the solution of the given problem types several biconjugate gradient-type methods for non-Hermitian systems are applicable. Their memory efficiency and their convergence properties allow for the solution of realistic, large scale problems as part of a general electromagnetic solver package as MAFIA [10] on modern desktop workstations.

2. Iterative Algorithms for Complex Symmetric Systems. To the solution of the complex linear system of equations

$$Ax = b \tag{1}$$

where A is non-Hermitian, but symmetric ($A = A^T \neq A^H$), quite often modern Krylov subspace methods such as the well-known CGS method [15] and its stabilized descendants TFQMR [5] and BiCGstab(l) [14], or the GMRES method [12], which are designed for more general non-Hermitian systems, are shown to be applicable (cf. [11], [19]). Most of these methods require more than one matrix vector multiplication per iteration step and do not exploit the given symmetry of the system matrix.

Methods which especially exploit the special structure of such matrices are usually of some minimal residual type as presented by Freund in [4], where methods for matrices of the type $A = e^{i\theta}(T + i\omega I)$ with Hermitian matrix T are derived, or are based on the complex version of the biorthogonal conjugate gradient (BiCG) method by Jacobs [9] on which the focus of this paper is set. With the complex inner product $\langle x, y \rangle := y^H x$ its vectorial formulation reads as:

BiCG algorithm

Choose x_0; $r_0 = b - Ax_0$; $p_0 = r_0$;
Choose $\tilde{r}_0$, such that $\langle r_0, \tilde{r}_0 \rangle \neq 0$; $\tilde{p}_0 = \tilde{r}_0$;
For $k = 0, 1, \ldots$ **do:**

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k p_k; \\ r_{k+1} &= r_k - \alpha_k A p_k; \qquad \tilde{r}_{k+1} = \tilde{r}_k - \bar{\alpha}_k A^H \tilde{p}_k; \\ p_{k+1} &= r_{k+1} + \beta_k p_k; \qquad \tilde{p}_{k+1} = \tilde{r}_{k+1} + \bar{\beta}_k \tilde{p}_k; \end{aligned}$$

where

$$\alpha_k = \frac{\langle r_k, \tilde{r}_k \rangle}{\langle A p_k, \tilde{p}_k \rangle}$$

$$\beta_k = \frac{\langle r_{k+1}, \tilde{r}_{k+1} \rangle}{\langle r_k, \tilde{r}_k \rangle}$$

When the complex linear system to be solved is also symmetric, it simplifies to a version where only one matrix vector multiplication per iteration step is needed. Since it has been republished by Melissen and van der Vorst [17] it is known as Conjugate Orthogonal Conjugate Gradient algorithm (COCG) and it coincides with the standard conjugate gradient (CG) algorithm for real-valued systems. The Quasi-Minimal Residual (QMR) method by Freund [6], which was derived for complex-symmetric linear systems, is closely connected to the COCG method (cf. [6], [22]).

With an extension of the approach used for the COCG method a whole class of BiCG-related algorithms for complex-symmetric systems can be derived which also require in the given formulation only one matrix vector multiplication per iteration step.

Let $\pi \in \Pi_n := \{\pi \mid \pi(z) = \sum_{i=0}^{n} c_i z^i, z \in \mathbf{C},\ c_i \in \mathbf{R}\ \forall\, i, c_n \neq 0\}, \Gamma := \{c_i \mid i = 0, \ldots, n\}$ and define

$$\tilde{r}_0 := \pi(\bar{A})\bar{r}_0, \tag{2}$$

where $\bar{A}$ and $\bar{r}_k$ correspond to the conjugate-complex of the matrix A and the vector r_k, respectively. Inserting this into the complex BiCG algorithm as given above it can be shown by induction that the following expressions

$$\tilde{r}_k := \pi(\bar{A})\bar{r}_k, \tag{3}$$

$$\tilde{p}_k := \pi(\bar{A})\bar{p}_k, \quad \forall\, k = 0, 1, \ldots \tag{4}$$

hold in the complex BiCG algorithm. Thus one achieves a class of BiCG-type algorithms for symmetric-complex linear systems of equations, SCBiCG(Γ, n), where each method of this class is specified by its set of polynomial coefficients Γ, and the number of the highest occuring matrix multiplicity $n+1$, which is implicitly already given in Γ.

To show that this class of algorithms requires only one matrix vector multiplication per iteration one defines the notation $v(i)_k := A^i v_k$ such that its formulation reads as

SCBiCG(Γ, n) algorithm

Choose x_0; $r(0)_0 = b - Ax_0$;
For $i = 0, \ldots$,n-1 **do:**
$r(i+1)_0 = Ar(i)_0$;
For $i = 0, \ldots$,n **do:**
$p(i)_0 = r(i)_0$;
$p(n+1)_0 = Ap(n)_0$;
For $k = 0, 1, \ldots$ **do:**

$$x_{k+1} = x_k + \alpha_k p(0)_k;$$

For $i = 0, \ldots$,n **do:**

$$r(i)_{k+1} = r(i)_k - \alpha_k p(i+1)_k;$$

For $i = 0, \ldots$,n **do:**

$$p(i)_{k+1} = r(i)_{k+1} + \beta_k p(i)_k;$$

$$p(n+1)_{k+1} = Ap(n)_{k+1};$$

with

$$\alpha_k = \frac{\sum\limits_{0 \leq l = i+j \leq n,\ j \leq i \leq j+1} c_l \cdot r(i)_k^T r(j)_k}{\sum\limits_{l=0}^{n} c_l \cdot p(l)_k^T p(1)_k}$$

$$\beta_k = \frac{\sum\limits_{0 \leq l = i+j \leq n,\ j \leq i \leq j+1} c_l \cdot r(i)_{k+1}^T r(j)_{k+1}}{\sum\limits_{0 \leq l = i+j \leq n,\ j \leq i \leq j+1} c_l \cdot r(i)_k^T r(j)_k}.$$

The iteration process of these methods is governed by the typical Petrov-Galerkin condition of the underlying BiCG algorithm. With the Krylov subspaces defined by $K_k = K_k(A, r_0) := \text{span}\{r_0, Ar_0, \ldots A^{k-1}r_0\}$ with $r_0 = b - Ax_0$ and $L_k = L_k(A^H, \tilde{r}_0)$, this condition reads as

$$b - Ax_k \perp L_k \text{ with } x_k \in x_0 + K_k. \tag{5}$$

For the iteration vectors x_k of an SCBiCG(Γ, n) method holds

$$\langle r_i, \pi(\bar{A})\bar{r}_j \rangle \quad = \quad r_j^T \pi(A) r_i = 0 \quad \forall\, 0 \le i, j \le k,\ i \ne j \tag{6}$$

and with $\chi(t) := t \cdot \pi(t) \in \Pi_{n+1}$

$$\langle Ap_i, \pi(\bar{A})\bar{p}_j \rangle \quad = \quad p_j^T \chi(A) p_i = 0 \quad \forall\, 0 \le i, j \le k,\ i \ne j. \tag{7}$$

In the practical implementations for this paper only polynomial degrees $n \in \{0, 1\}$ are considered. The choice of $n = 0$ directly yields the aforementioned COCG algorithm for complex symmetric systems [17], which coincides with the standard CG algorithm for real valued systems. The case $n = 1$ with $c_0 = 0$ results in a method which coincides for real-valued systems with the Conjugate Residual method (CR) of Stiefel [16] and to which will be referred to as BiCGCR method in the following. The CR method is mathematically equivalent to the GMRES method [12] for Hermitian systems $A = A^H$ and has a residual minimization property in the complex Euclidean norm. In [3] it was shown that for complex symmetric linear systems it is more natural to consider the complex bilinear form $(x, y) := y^T x$ as the governing 'inner product' (also cf. [6]). For BiCGCR the identity

$$\beta_k \quad = \quad -\frac{r_{k+1}^T A^2 p_k}{p_k^T A^2 p_k} = \frac{r_{k+1}^T A r_{k+1}}{r_k A r_k} \tag{8}$$

holds and from this a residual minimization property with respect to the complex bilinear form (.,.) can easily be shown to hold for the BiCGCR algorithm:

$$\beta_k = \arg\, min_{\beta \in \mathbf{C}} (r_k - \beta A p_k, r_k - \beta A p_k)^{\frac{1}{2}}. \tag{9}$$

Unless the linear system has not just vanishing imaginary parts, i.e., the system has simplified to the real-symmetric case, this special type of 'minimal' residual projection does not transfer to the complex Euclidean residual norm, with which the convergence of the iteration process is usually monitored. Here the usual oscillations which are typical for the BiCG iteration process are to be expected and could be observed in all the numerical examples.

Especially for the complex symmetric matrices of the type $A = A_1 + \omega D$, where the non-Hermitian part is introduced by a complex diagonal matrix D multiplied by a real scalar ω and where A_1 is real symmetric and indefinite, the aforementioned residual minimization with respect to the complex bilinear form (.,.) seems to have a favourable effect in practice, since the oscillations of the BiCGCR method seem to be less extreme than in the COCG method.. This property depends on the magnitude of ω, i.e., the difference of magnitudes in the real and imaginary parts of the matrix entries of A. To avoid strong oscillations in the residual norm is desirable with respect to the result on the loss of accuracy of BiCG-type iteration processes given in [14]

$$| \, \|r_k\| - \|b - Ax_k\| \, | \le \bar{\zeta} k n_A \, \| \, |A| \, \| \, \left\|A^{-1}\right\| \max_{0 \le j \le k} \|r_j\|, \tag{10}$$

where $\|\cdot\|$ is the Euclidean norm, $\bar{\zeta}$ is the machine precision and n_A is the maximum number of non-zero entries per row of A.

All the presented methods try to solve the complete complex-valued problem. A different approach is applied by [1] using a special decomposition of the matrix into real and imaginary part, where the resulting real-valued linear system is solved iteratively.

Smoothing of the residuals. In [22] the minimal residual smoothing (MRS) technique for iterative methods developed by Schönauer [13] is presented for real systems, its extension to complex linear system solvers is straightforward: Given an iteration method producing the vector iterates x_k of approximate solutions

to the linear system and r_k a sequence of the related residuals, two auxiliary vector sequences y_k and s_k can be introduced as follows:

$$y_0 := x_0, \quad s_0 := r_0, \tag{11}$$

$$y_k := (1-\eta_k)y_{k-1} + \eta_k x_k, \quad s_k := (1-\eta_k)s_{k-1} + \eta_k r_k \quad \text{for } k = 1, 2, \ldots \tag{12}$$

and $\eta_k \in \mathbf{R}$ chosen such that $\|b - Ay_k\|_2$ is minimized in each iteration step k. The choice of

$$\eta_k := -\frac{s_{k-1}^H(r_k - s_{k-1}) + (r_k - s_{k-1})^H s_{k-1}}{2 \cdot \|r_k - s_{k-1}\|_2^2} \tag{13}$$

for the complex systems yields the desired residual minimization property. The connection between smoothing and the quasi-minimal residual smoothing introduced by Freund and Nachtigal [7] was extensively analysed in [22] and [21]. Numerical errors causing $b - Ay_k$ and s_k to differ too much from each other can be avoided by a mathematically equivalent reformulation of the underlying iteration schemes given in [22] .

3. The Finite Integration Theory for Maxwell's Equations and Numerical Results. The Finite Integration Theory is a representation of Maxwell's Equations in their integral form on a doublet of staggered orthogonal grids $G - \tilde{G}$ and may be considered as a finite-volume discretization approach especially suited for Maxwell's Equations. The resulting Maxwell Grid Equations preserve analytical and algebraic properties that ensure accurate numerical results and enable an algebraically exact self-testing of numerical results. The Finite Integration Theory and its notation is described in [20]. For the calculations here electro-quasistatic and driven frequency domain problems are of special interest. The background of electro-quasistatics is the modeling of arc-overs on contaminated insulators in high-voltage power plants. A thorough description of the numerical modeling is given in [19]. The electro-quasistatic fields can be determined by solving the complex potential problem,

$$\text{div } ((i\omega\varepsilon + \sigma) \text{ grad } \underline{\varphi}) = \text{ div } (\underline{\vec{J}}_0), \tag{14}$$

where the application of FIT yields a linear system

$$(A_\sigma + i\omega A_\varepsilon)\,\underline{\Phi}_E = \underline{p}_0. \tag{15}$$

In the matrix $A = A_\sigma + i\omega A_\varepsilon$ the matrix A_σ is related to the stationary currents and A_ε is the matrix of electrostatics, scaled with frequency ω. Both are symmetric positive definite, sparse and banded, such that A is a sparse matrix connecting neighbouring potentials. One typical small example calculation for this type of system is a simple contaminated square plate capacitor model with side length of 5 cm and a thickness of 4 cm. Its dielectric material is assumed to have a relative permittivity of $\varepsilon_r = 4$ and a conductivity of $\sigma = 10^{-12}$ S/m. The contamination is a layer of water on one side of the capacitor with a relative permittivity $\varepsilon_r = 81$ and a conductivity of $\sigma = 10^{-10}$ S/m. A voltage gradient of 15 kV/cm is assumed at a frequency of 50 Hz. The rank of the resulting complex symmetric linear system to be solved is 41616. The convergence curves in Fig. 1 show the convergence history of the non-preconditioned solvers. Note that the convergence behaviour of QMR, the minimal residual smoothing curve of COCG and of BiCGCR are in this case almost identical with slight advantages for BiCGCR. A more realistic example of an application was is the simulation of a cylindrical resonator of epoxy resin which is modelled in a $57 \times 57 \times 73$ grid yielding a system of 237177 complex variables. Some results were already published in [18].

The frequency domain solver W3 [8], in which the COCG method has been successfully used for several years now, has been enhanced for test purposes with a minimal residual smoothing (MRS), with the BiCGCR method with MR smoothing and the symmetric-complex QMR method. Their results could be compared with the performance of the TFQMR method which was earlier implemented for non-symmetric complex linear systems, as they occur, when special waveguide boundaries are applied to the computational region [2]. In the TFQMR algorithm as given in [5] only an upper norm bound is generated for the residuals. Additionally for comparison the true residuals are evaluated in the present implementation every 100 iteration steps. The calculations of driven problems are based on the so-called *curl-curl*-equation [8] for the electric field

$$\text{curl } \frac{1}{\mu} \text{ curl } \vec{E} - \omega^2 \varepsilon \vec{E} = -i\omega \vec{J}_0 \tag{16}$$

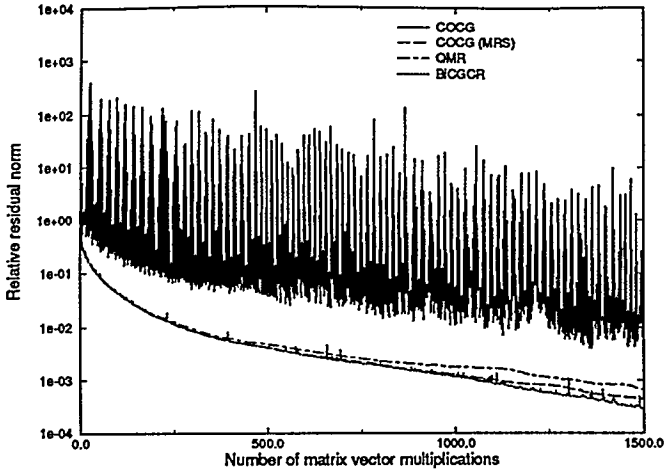


Figure 1: Example 1: Convergence history of an example calculation in electro-quasistatics with 41616 unknowns. No preconditioning is used. Note the wild oscillations of COCG in comparison to the almost monotone convergence of BICGCR which is the fastest method.

which results via FIT in a system matrix

$$A = A_1 - \omega^2 D, \tag{17}$$

where A_1 is an indefinite, real (for real-valued μ), symmetric matrix. D is a diagonal matrix, which has complex entries in the general lossy case, such that A is complex symmetric. Note that frequencies ω close to the eigenfrequencies of A_1 result in an almost singular system matrix A. The following example shows the convergence curves from a simulation of a short antenna near a lossy cube. The cube has a conductivity $\sigma = 0.5$ S/m and a relative permittivity $\varepsilon_r = 3$. The antenna is driven with 150 MHz close to the cube. The example in Fig. 2 is discretized with 60000 unknowns.

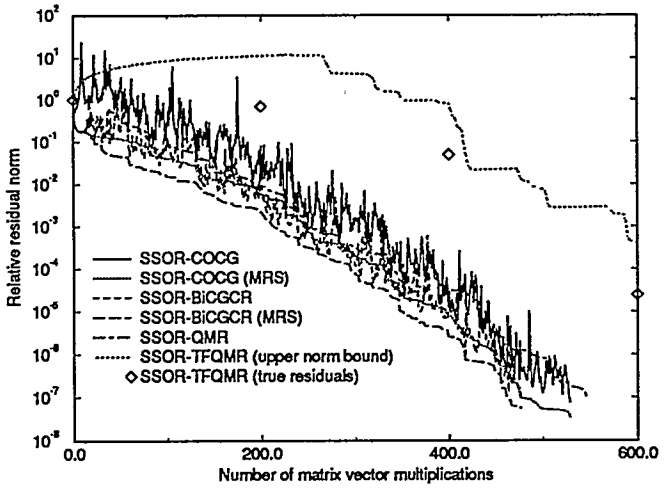


Figure 2: Convergence history of an example of a driven frequency domain problem with 60000 complex unknowns. SSOR preconditioning is used. The TFQMR method for general non-Hermitian problems here is not competitive in its speed of convergence. The MRS iteration vectors of the BiCGCR process show the fastest convergence.

A quarter of a transmitter structure, where the lossy material is modelled with values $\sigma = 0.25$ S/m and $\varepsilon_r = 1$, is excited by a surrounding antenna at a frequency of 1 Mhz. It is discretized with 10625 complex unknowns. Calculation shows that a shift in the SCBiCG($\Gamma = \{c_0, c_1\}, 1$) methods from the COCG method

to the BiCGCR method introduced by the condition $c_0 + c_1 = 1$ on the related sets of polynomial coefficients Γ is also represented in the correspondent convergence processes (cf. Fig. 3).

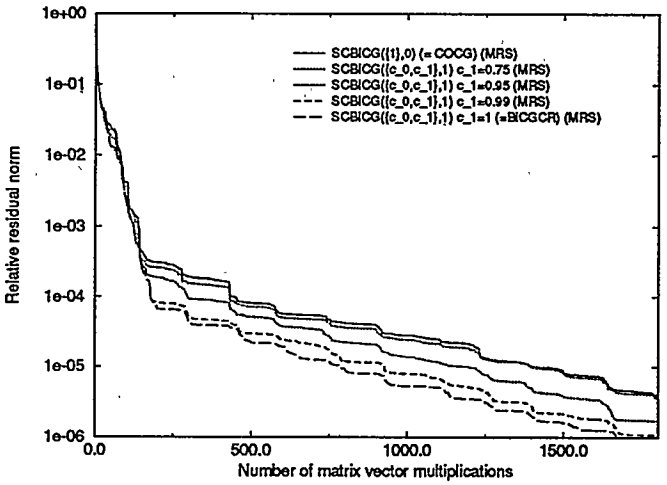


Figure 3: Convergence history for several SCBiCG(Γ, 1) methods with MRS, where the coefficients $c_0, c_1 \in \Gamma$ fulfill $c_0 + c_1 = 1$. Note that the BiCGCR method exhibits the fastest convergence of all methods.

All the numerical tests of the algorithms presented above were performed on practical 3D examples from electro-quasistatics in the static solver module S and driven frequency domain computations with the frequency domain solver module W3 [8] of MAFIA [10]. They were carried out with both non-preconditioned and with memory efficient SSOR-preconditioned systems. For all but the quasi-minimal residual methods a minimal residual smoothing was applied. The computations have been performed on a SUN Sparc20 workstation.

4. Concluding Remarks. The SCBiCG class of algorithms is a direct extension of the idea of the derivation of the COCG method from the complex BiCG algorithm. For this paper the COCG and the BiCGCR method which correspond to SCBiCG($\{1\}, 0$) and SCBiCG($\{0, 1\}, 1$) were implemented and used in test calculations. In connection with Schönauer's minimal residual smoothing for the complex case both methods work properly on practical problems arising from two special problem types in electromagnetic field calculation. Of course, theoretically BiCG-related breakdown or near-breakdown of the algorithms due to the failing or inaccurate calculation of the BiCG-coefficients is always possible in these methods. However, a catastrophic breakdown has never been encountered in all our practical calculations.

5. Acknowledgements. The authors would like to thank Peter Hahne and Rolf Schuhmann for their helpful suggestions on this paper and for the friendly discussions on its content.

References

[1] O. Axelsson and A. A. Kutcherov. Real valued iterative methods for solving complex linear systems. Dept. of Mathematics and Informatics, University Nijmegen; unpublished, 1996.

[2] M. Clemens, R. Schuhmann, U. sula van Rienen, and T. Weiland. Modern Krylov subspace methods in electromagnetic field computation using the Finite Integration Theory. *ACES Journal, Special Issue on Applied Mathematics: Meeting the challenges presented by Computational Electromagnetics*, 11(1):70–84, March 1996.

[3] B. D. Craven. Complex symmetric matrices. *J. Austral. Math. Soc.*, 10:341–354, 1969.

[4] R. W. Freund. On conjugate gradient methods and polynomial preconditioners for a class of complex non-Hermitian matrices. *Numer. Math.*, 57:285–312, March 1990.

[5] R. W. Freund. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comput.*, 14(2):470–482, March 1993.

[6] R. W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Stat. Comp.*, 13(1):425–448, January 1994.

[7] R. W. Freund and N. Nachtigal. QMR: A quasi-minimal residual method for non-Hermitian matrices. *Numer.Math.*, 60:315–339, 1991.

[8] P. Hahne. *Zur Numerischen Berechnung Zeitharmonischer Elektromagnetischer Felder.* PhD thesis, Technische Universität Darmstadt, 1992.

[9] D. A. H. Jacobs. *The Exploitation of Sparsity by Iterative Methods.* Sparse matrices and their Uses. I. S. Duff, Springer, 1981. pp. 191-222.

[10] The Mafia collaboration. *User's Guide MAFIA Version 3.x.* CST GmbH, Lauteschlägerstr.38, D-64289 Darmstadt.

[11] M. Reissel. 3D eddy-current computation using Krylov subspace methods. Technical Report 94, Fachbereich Mathematik, Universität Kaiserlautern, August 1993.

[12] Y. Saad and M. H. Schultz. GMRES: A generalised minimal residual method for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[13] W. Schönauer. Scientific computing on vector computers. North-Holland, Amsterdam, New-York, Oxford, Tokio, 1987.

[14] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema. BICGSTAB(l) and other hybrid Bi-CG methods. *Numerical Algorithms*, 7:75–109, 1994.

[15] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10(1):36–52, November 1989.

[16] E. Stiefel. Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme. *Comment. Math. Helv.*, 29:157–179, 1955.

[17] H. van der Vorst and J. Melissen. A Petrow-Galerkin type method for solving Ax=b, where A is symmetric complex. *IEEE Trans.Mag.*, 26(2):706–708, 1990.

[18] U. van Rienen, M. Clemens, and T. Weiland. Computation of low-frequency electromagnetic fields. In *Proceedings of the ICIAM95 Conference, Berlin*, 1995. To appear in ZAMM.

[19] U. van Rienen, M. Clemens, and T. Weiland. Simulation of low-frequency fields on high-tension insulators with light contaminations. In *Proceedings of the Compumag95 Conference, Berlin*, 1995. To appear in IEEE Transactions on Magnetics, May 1996.

[20] T. Weiland. On the unique numerical solution of Maxwellian eigenvalue problems in three dimensions. *Particle Accelerators*, 17:227–242, 1985.

[21] R. Weiss. Relations between smoothing and QMR. Interner Bericht 53/94, Rechenzentrum der Universität Karlsruhe, January 1994.

[22] L. Zhuo and F. Walker. Residual smoothing techniques for iterative methods. *SIAM J. Sci. Comput.*, 15(2):297–312, March 1994.

Matrix Equation Decomposition and Parallel Solution of Systems Resulting from Unstructured Finite Element Problems in Electromagnetics

Tom Cwik* and Daniel S. Katz**

*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

**Cray Research, 222 N. Sepulveda Blvd., Ste. 1406, El Segundo, CA 90245

Finite element modeling has proven useful for accurately simulating scattered or radiated electromagnetic fields from complex three-dimensional objects whose geometry varies on the scale of a fraction of an electrical wavelength. An unstructured finite element model of realistic objects leads to a large, sparse, system of equations that needs to be solved efficiently with regard to machine memory and execution time. Both factorization and iterative solvers can be used to produce solutions to these systems of equations. Factorization leads to high memory requirements that limit the electrical problem size of three-dimensional objects that can be modeled. An iterative solver can be used to efficiently solve the system without excessive memory use and in a minimal amount of time if the convergence rate is controlled.

This paper will discuss a number of topics related to the parallel creation and solution of matrices resulting from large unstructured problems, in the context of an electromagnetic finite element code running on the Cray T3D located at the Jet Propulsion Laboratory. The JPL code, named PHOEBUS, has been used to obtain solutions for systems with nearly three-quarters of a million unknowns.

One of these topics is mesh vs. matrix partitioning. The code running at JPL decomposes the finite element matrix in row slabs, as compared with the usual strategy of decomposing the mesh. Another topic is the iterative solver, in this case a quasi minimum residual method. An examination of the computational kernel, a sparse matrix dense vector multiply, is given, and the issue of solving for a single right hand side vs. multiple right hand sides (possibly a block of right hand sides) is discussed. Another question related to the iterative solver is parallel methods of preconditioning, such as using an incomplete Cholesky factorization or a sparse approximate inverse.

Iterative Solution of the Semiconductor Device Equations

S.W. Bova & G.F. Carey
The Computational Fluid Dynamics Laboratory
The University of Texas at Austin

Abstract

Most semiconductor device models can be described by a nonlinear Poisson equation for the electrostatic potential coupled to a system of convection-reaction-diffusion equations for the transport of charge and energy. These equations are typically solved in a decoupled fashion and *e.g.* Newton's method is used to obtain the resulting sequences of linear systems. The Poisson problem leads to a symmetric, positive definite system which we solve iteratively using conjugate gradient. The transport equations lead to nonsymmetric, indefinite systems, thereby complicating the selection of an appropriate iterative method. Moreover, their solutions exibit steep layers and are subject to numerical oscillations and instabilities if standard Galerkin-type discretization strategies are used. In the present study, we use an upwind finite element technique for the transport equations. We also evaluate the performance of different iterative methods for the transport equations and investigate various preconditioners for a few generalized gradient methods. Numerical examples are given for a representative two-dimensional depletion MOSFET.

Most semiconductor device models can be described by a nonlinear Poisson equation for the electrostatic potential, ψ, coupled with a system of convection-diffusion-reaction equations for the transport of charge and energy [1]. More specifically, ψ satifies an equation of the form

$$\nabla^2\psi = \frac{q}{\varepsilon}(n - p - C), \tag{1}$$

where q is the electron charge, ε is the permittivity of the semiconductor, n and p are the electron and hole concentrations, and C denotes the given impurity distribution.

The remaining conservation relations lead to a coupled set of transport equations for n, p, and energy. These deterministic transport models can be broadly classified as drift-diffusion (DD) or hydrodynamic (HD). Drift-diffusion models can be derived from HD ones by making certain assumptions, such as neglecting changes in the charge carrier energies. Hydrodynamic models treat the carrier energy as an unknown and so contain additional convection–diffusion–reaction equations. In the present study, we consider only the DD equations in order to obtain representative nonsymmetric matrix problems in the fully discrete model.

The charge concentrations satisfy

$$-\nabla \cdot \mathbf{J}_n = -qR(\psi, n, p) \tag{2}$$

$$\nabla \cdot \mathbf{J}_p = -qR(\psi, n, p) \tag{3}$$

where $\mathbf{J}_n$ and $\mathbf{J}_p$ denote the electron and hole current densities, and $R(\psi, n, p)$ is the net recombination/generation rate. The most widely used constitutive relations for current densities are modeled as a combination of a drift (convection) term and a diffusion term, and so may be written as

$$\mathbf{J}_n = -q\mu_n \nabla\psi n + qD_n \nabla n \tag{4}$$

$$\mathbf{J}_p = -q\mu_p \nabla\psi p - qD_p \nabla p \tag{5}$$

where μ_n and $\mu_p$denote the electron and hole mobilities, and D_n, D_p are the corresponding diffusivities.

For certain classes of devices such as silicon MOSFETS, it is often assumed that one current density, *e.g.* $\mathbf{J}_p$, is negligible. Then we may express p in terms of the Fermi potential, Φ_p, namely

$$p = n_i \exp\left[\frac{q}{k_B T}(\Phi_p - \psi)\right] \tag{6}$$

where n_i is the known intrinsic concentration of the semiconductor, k_B is Boltzmann's constant, and T is temperature of the semiconductor lattice.

Upon substitution of (4) into (2), we obtain the convection–diffusion–reaction equation for transport of electrons

$$\nabla \cdot (\mu \nabla\psi n) - \nabla \cdot (D \nabla n) = -R(\psi, n, p), \tag{7}$$

where we have dropped the subscript $()_n$. In a typical model, μ is specified using empirical functions of ψ, C, and J, and D is given by Einstein's relation, $D = \mu k_B T/q$. To complete the model, we assume that only one recombination/generation mechanism is present, namely Shockley–Read–Hall, so that

$$R(\psi, n, p) = \frac{np - n_i^2}{\tau_p(n + n_i) + \tau_n(p + n_i)} \tag{8}$$

where τ_p and τ_n are the hole and electron recombination lifetimes.

Equations (1) and (7) consititute a pair of coupled, nonlinear partial differential equations for ψ and n. Gummel's method [2] is the most common approach to solving this system. This decoupled strategy begins by linearizing (1). Then starting iterates $\psi^{(0)}$ and $n^{(0)}$ are used to obtain a new estimate, $\psi^{(1)}$. In turn, this improved estimate is used in (7) to generate $n^{(1)}$. Each equation is alternately satisfied until the procedure converges.

Let us first consider the initial linearization step. This is accomplished by introducing an iteration level k into (1):

$$\nabla^2 \psi^{(k+1)} = \frac{q}{\varepsilon}(n^{(k+1)} - p^{(k+1)} - C), \tag{9}$$

Now, linearize $n^{(k+1)}$ according to $n^{(k+1)} = n^{(k)} + \delta n$, where $\delta n = \frac{\partial n}{\partial \psi}\delta\psi$ and $\frac{\partial n}{\partial \psi}$ is computed from the quasi-Fermi potential relation

$$n = n_i \exp\left[\frac{q}{k_B T}(\psi - \phi_n)\right], \tag{10}$$

where ϕ_n is the electron quasi–Fermi potential. A similar manipulation for $p^{(k+1)}$ leads to the decoupled, linearized potential equation

$$\nabla^2 \psi^{(k+1)} - \frac{q}{\varepsilon}(n^{(k)} + p^{(k)})\psi^{(k+1)} = \frac{q}{\varepsilon}\left[(n - p - C - \psi(n + p))\right]^{(k)}. \tag{11}$$

We discretize (11) using Galerkin's method on linear triangles and are led to the sparse, symmetric, positive definite system

$$\mathbf{A}_\psi \psi^{(k+1)} = \mathbf{b}_\psi \tag{12}$$

which we solve using conjugate gradients (CG) as described later.

Using the solution $\psi^{(k+1)}$ of (12), in the Gummel strategy we next update the carrier concentration using the transport equation. Introducing a Picard iteration, (7) becomes

$$\nabla \cdot (\mu_l \nabla \psi^{(k+1)} n_{l+1}) - \nabla \cdot (D_l \nabla n_{l+1}) = -R(n_l, \psi^{(k+1)}), \tag{13}$$

where n_{l+1} is the new carrier concentration iterate to be completed.

As is well known, Galerkin's method often fails when used to discretize convection-diffusion equations. Hence we use an SUPG type Petrov-Galerkin strategy (*e.g.* see [3]) to discretize (13). We are accordingly led to a sequence of nonsymmetric, linear systems which may be written as

$$\mathbf{A}_n \mathbf{n}_{l+1} = \mathbf{b}_n. \tag{14}$$

Summarizing, Gummel's method for the semiconductor device problem with generalized gradient iterative solution of the linear algebraic subsystems is:

Obtain $\psi^{(0)}, n^{(0)}$
$k = 0$
do while(not finished)
  solve (12) for $\psi^{(k+1)}$ via preconditioned CG
  $n_0 = n^{(k)}, l = 0$
  do while(not finished)
    solve (14) for n_{l+1} via preconditioned bi-CG, *etc.*
    $l = l + 1$
  end do
  $n^{(k+1)} = n_l$
  $k = k + 1$
end do

In the present work, we study the use of several of the iterative strategies in the package NSPCG [4] to solve the systems (12) and (14). More specifically, for the SPD system (12), we use conjugate gradient and evaluate the performance of Richardson, Jacobi, incomplete Cholesky, and the polynomial preconditioners Neumann and least-squares. For the nonsymmetric system (14), we consider Richardson, Jacobi, incomplete LU factorization and Neumann polynomial preconditioners in conjunction with three methods: biconjugate gradient (BCG), conjugate gradient squared (CGS), and stabilized conjugate gradients (CGSTAB). The purpose of this study is to investigate the behavior of these iterative methods and to obtain experimental evidence as to which, if any, of the these iterative strategies is most effective for this class of problems.

First, we consider the solution of the potential equation. For a silicon MOSFET under applied biases in the subthreshold region, the electron Fermi potential can also be specifed and the concentration equation

is thereby eliminated. We are then left with a single equation which may be solved for an approximation to ψ. Let us consider the MOSFET configuration described in [5, 6] with gate and drain voltages of $V_g = 0.1$V and $V_d = 0.4$V, respectively. The starting iterate is the most recent approximation to ψ obtained for $V_g = 0.1$V and $V_d = 0.35$V. Our mesh has 3,046 nodes and 5,953 triangles. All computations which we report were performed in double precision on a DEC Alpha workstation. We observed in this experiment that the L^2 norm of the nonlinear residual was reduced by 12 orders of magnitude after 11 nonlinear iteration steps. We also noted that the Richardson and modified incomplete Cholesky preconditioners failed. Even though it requires more CG iterations, Jacobi preconditioning was the least expensive in terms of CPU time expended because it is computationally an inexpensive preconditioner to evaluate. The next two best–perfoming preconditioners were first–order Neumann and first–order least–squares, respectively.

Next, we use the DD model to simulate the same MOSFET at bias conditions of $V_d = 3$V and $V_g = 2$V. The finite element mesh used in this study has 5,593 nodes. In this case, we found CGS to be less robust, in the sense that it failed when combined with the Richardson and first and third–order Neumann preconditioners. In contrast, BCG and CGSTAB converged with these preconditioners, as well as with Jacobi, incomplete LU factorization, and first and fourth–order Neumann. As with the potential equation, all three nonsymmetric iterative methods converged in the least amount of CPU time when used in conjunction with Jacobi preconditioning. Finally, we note that CGSTAB was observed to usually require less than half the time of the other two nonsymmetric solvers. In the talk, we discuss these issues and others, such as iteration counts and nonlinear convergence rates.

Acknowledgement

This research was sponsored by National Science Foundation grants ASC-9405084 and ECS-9412475.

References

[1] M. Lundstrom, *Fundamentals of Carrier Transport*, Addison-Wesley, Reading, MA, 1990.

[2] H.K. Gummel, "A Self-Consistent Iterative Scheme for One-Dimensional Steady State Transistor Calculations", *IEEE Transactions on Electron Devices*, vol. ED-11, pp. 455-465, 1964.

[3] Hughes, T.J.R., Mallet, M., and Mizukami, A., "A New Finite Element Formulation for Computational Fluid Dynamics: II. Beyond SUPG", *Computer Methods in Applied Mechanics and Engineering*, 54, pp. 341-355, 1986.

[4] Oppe, T.C., Joubert, W.D., and Kincaid, D.R., "NSPCG User's Guide: A Package for Solving Large Sparse Linear Systems by Various Iterative Methods," Center for Numerical Analysis Report CNA-216, The University of Texas at Austin, April, 1988.

[5] Greenfield, J.A. and Dutton, R.W., "Analysis of Nonplanar Devices", *NATO Advanced Study Institute on Process and Device Simulation for MOS-VLSI Circuits*, Boston, 1983.

[6] Sharma, M.S., *Finite Element Modelling of Semiconductor Devices*, Ph.D. Dissertation, The University of Texas at Austin, December, 1988.

Topic: *Multiple RHS*	*Session Chair:* *Roland Freund*	*Room C*
8:00 - 8:30	W. Boyse	Multiple Solutions to Dense Systems in Radar Scattering using a Preconditioned Block GMRES Solver
8:30 - 9:00	R. Freund	The BL-QMR Algorithm for Non-Hermitian Linear Systems with Multiple Right-Hand Sides
9:00 - 9:30	M. Malhotra	Iterative Solution of Multiple Radiation and Scattering Problems in Structural Acoustics Using the BL-QMR Algorithm
9:30 - 10:00	T. Chan	Galerkin Projection Methods for Solving Multiple Related Linear Systems

Multiple Solutions to Dense Systems in Radar Scattering using a Preconditioned Block GMRES Solver

William E. Boyse
Advanced Software Resources, Inc.
3375 Scott Boulevard, Suite 420
Santa Clara, CA 95054
Boyse@netcom.com

Multiple right-hand sides occur in radar scattering calculations in the computation of the simulated radar return from a body at a large number of angles. Each desired angle requires a right-hand side vector to be computed and the solution generated. These right-hand sides are naturally smooth functions of the angle parameters and this property is utilized in a novel way to compute solutions an order of magnitude faster than LINPACK.

The modeling technique addressed is the Method of Moments (MOM), i.e a boundary element method for time harmonic Maxwell's equations. Discretization by this method produces general complex dense systems of rank 100's to 100,000's. The usual way to produce the required multiple solutions is via LU factorization and solution routines such as found in LINPACK.

Our method uses the block GMRES iterative method to directly iterate a subset of the desired solutions to convergence. The computed Krylov sub-space is then used to approximate solutions to the remaining right-hand sides. We use the property that the right-hand sides are smooth functions of angle to select the subset of right-hand sides to directly iterate that will provide for optimal solution times.

The convergence rate of the GMRES algorithm is enhanced both by the block algorithm and by preconditioning the system with a sparse incomplete LU factorization generated by an ILU(T) algorithm with partial pivoting. This "drop tolerance" algorithm permits adjustment of the completeness of the preconditioner and guidelines are shown for choosing the most beneficial completeness percentage.

This algorithm is analyzed on a realistic MOM matrix problem of rank 3,400 to determine the parameters for efficient operation. The error in computed RCS values, the equation residual error and solution error are all used in this analysis. A method, based on Nyquist sampling theory, is shown to predict an optimal method of choosing the angles to directly iterate such that the total solution process is most efficient. It is shown in this case that the solver runs nearly an order of magnitude faster than single precision LINPACK.

The BL-QMR Algorithm for Non-Hermitian Linear Systems With Multiple Right-Hand Sides

Roland W. Freund
AT&T Bell Laboratories
Room 2C–420
600 Mountain Avenue
Murray Hill, New Jersey 07974–0636
email: freund@research.att.com

Many applications require the solution of multiple linear systems that have the same coefficient matrix, but differ in their right-hand sides. Instead of applying an iterative method to each of these systems individually, it is potentially much more efficient to employ a block version of the method that generates iterates for all the systems simultaneously. However, it is quite intricate to develop robust and efficient block iterative methods. In particular, a key issue in the design of block iterative methods is the need for deflation. The iterates for the different systems that are produced by a block method will, in general, converge at different stages of the block iteration. An efficient and robust block method needs to be able to detect and then deflate converged systems. Each such deflation reduces the block size, and thus the block method needs to be able to handle varying block sizes. For block Krylov-subspace methods, deflation is also crucial in order to delete linearly and almost linearly dependent vectors in the underlying block Krylov sequences. An added difficulty arises for Lanczos-type block methods for non-Hermitian systems, since they involve two different block Krylov sequences. In these methods, deflation can now occur independently in both sequences, and consequently, the block sizes in the two sequences may become different in the course of the iteration, even though they were identical at the beginning.

In this talk, we present a block version of Freund and Nachtigal's quasi-minimal residual (QMR) method for the solution of non-Hermitian linear systems with single right-hand sides. The QMR algorithm is a Krylov-subspace iteration that uses a look-ahead variant of the classical nonsymmetric Lanczos process to build basis vectors for the underlying Krylov subspaces and generates QMR iterates defined by a quasi-minimization of the residual norm. The block-QMR method (referred to as BL-QMR hereafter) is an extension of QMR to multiple linear systems. The BL-QMR method uses a novel Lanczos-type process for multiple starting vectors, which was recently developed by Aliaga, Boley, Freund, and Hernández, to compute suitable basis vectors for the two underlying block Krylov subspaces. The BL-QMR iterates are characterized by a block version of the quasi-minimization property, which can be formulated as a matrix least-squares problem. The underlying Lanczos-type process can handle the most general case of block Krylov sequences with arbitrary block sizes, and in particular, can also handle deflation in both sequences. The BL-QMR method employs the deflation procedure of the Lanczos-type process to detect and delete linearly and almost linearly dependent vectors in the underlying block Krylov sequences. In addition, BL-QMR also includes a deflation procedure to identify and drop linear systems whose solution can be recovered from the solutions of the remaining multiple linear systems. First, we describe the basic BL-QMR method and some of its important implementation details. We then present some theoretical properties of BL-QMR, such as error bounds. Finally, we report numerical results that illustrate typical features of the block-QMR method.

This is joint work with Manish Malhotra (Stanford University).

Iterative Solution of Multiple Radiation and Scattering Problems in Structural Acoustics Using the BL-QMR Algorithm

Manish Malhotra
Department of Civil Engineering
Stanford University
Stanford, California 94305–4020
email: manish@am-sun2.stanford.edu

Finite-element discretizations of time-harmonic acoustic wave problems in exterior domains result in large sparse systems of linear equations with complex symmetric coefficient matrices. In many situations, these matrix problems need to be solved repeatedly for different right-hand sides, but with the same coefficient matrix. For instance, multiple right-hand sides arise in radiation problems due to multiple load cases, and also in scattering problems when multiple angles of incidence of an incoming plane wave need to be considered.

In this talk, we discuss the iterative solution of multiple linear systems arising in radiation and scattering problems in structural acoustics by means of a complex symmetric variant of the BL-QMR method. First, we summarize the governing partial differential equations for time-harmonic structural acoustics, the finite-element discretization of these equations, and the resulting complex symmetric matrix problem. Next, we sketch the special version of BL-QMR method that exploits complex symmetry, and we describe the preconditioners we have used in conjunction with BL-QMR. Finally, we report some typical results of our extensive numerical tests to illustrate the typical convergence behavior of BL-QMR method for multiple radiation and scattering problems in structural acoustics, to identify appropriate preconditioners for these problems, and to demonstrate the importance of deflation in block Krylov-subspace methods. Our numerical results show that the multiple systems arising in structural acoustics can be solved very efficiently with the preconditioned BL-QMR method. In fact, for multiple systems with up to 40 and more different right-hand sides we get consistent and significant speed-ups over solving the systems individually.

This is joint work with Roland W. Freund (AT&T Bell Laboratories).

GALERKIN PROJECTION METHODS FOR SOLVING MULTIPLE RELATED LINEAR SYSTEMS

TONY F. CHAN , MICHAEL NG , AND W. L. WAN

We consider using Galerkin projection methods for solving multiple related linear systems $A^{(i)}x^{(i)} = b^{(i)}$ for $1 \leq i \leq s$, where $A^{(i)}$ and $b^{(i)}$ are different in general. We start with the special case where $A^{(i)} = A$ and A is symmetric positive definite. The method generates a Krylov subspace from a set of direction vectors obtained by solving one of the systems, called the seed system, by the CG method and then projects the residuals of other systems orthogonally onto the generated Krylov subspace to get the approximate solutions. The whole process is repeated with another unsolved system as a seed until all the systems are solved. We observe in practice a super-convergence behaviour of the CG process of the seed system when compared with the usual CG process. We also observe that only a small number of restarts is required to solve all the systems if the right-hand sides are close to each other. These two features together make the method particularly effective. In this talk, we give theoretical proof to justify these observations. Furthermore, we combine the advantages of this method and the block CG method and propose a block extension of this single seed method.

The above procedure can actually be modified for solving multiple linear systems $A^{(i)}x^{(i)} = b^{(i)}$, where $A^{(i)}$ are now different. We can also extend the previous analytical results to this more general case. Applications of this method to multiple related linear systems arising from image restoration and recursive least squares computations are considered as examples.

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*	*Room A*
10:30 - 11:00	R. Hornung	Adaptive Mesh Refinement and Multilevel Iteration for Multiphase, Multicomponenet Flow in Porous Media
11:00 - 11:30	J. Jones	Semi-Coarsening Multigrid Methods for Parallel Computing
11:30 - 12:00	P. Vanek	An Algebraic Multigrid Algorithm for Symmetric Positive Definite Linear Systems

Adaptive Mesh Refinement and Multilevel Iteration for Multiphase, Multicomponent Flow in Porous Media

Richard D. Hornung
Department of Mathematics, Duke University
Durham, NC 27708-0320
e-mail: hornung@math.duke.edu

Abstract

An adaptive local mesh refinement (AMR) algorithm originally developed for unsteady gas dynamics is extended to multi-phase flow in porous media. Within the AMR framework, we combine specialized numerical methods to treat the different aspects of the partial differential equations. Multi-level iteration and domain decomposition techniques are incorporated to accommodate elliptic/parabolic behavior. High-resolution shock capturing schemes are used in the time integration of the hyperbolic mass conservation equations. When combined with AMR, these numerical schemes provide high resolution locally in a more efficient manner than if they were applied on a uniformly fine computational mesh. We will discuss the interplay of physical, mathematical, and numerical concerns in the application of adaptive mesh refinement to flow in porous media problems of practical interest.

Multi-phase flow in oil recovery and aquifer remediation problems exhibits behavior typical of the solutions to both elliptic/parabolic and hyperbolic partial differential equations. For example, pressure changes are felt quickly throughout the reservoir when the flow is incompressible or only slightly compressible. In contrast, injected fluids move with finite speeds and the flow can develop sharp fronts separating different fluid states. Generally, the equations of multi-phase porous media flow can be written as a system of conservation equations for the masses of the fluid components, subject to a constraint that the fluid fills the void space in the rock. If the model describes incompressible flow, one obtains an elliptic pressure equation by summing the mass conservation equations and applying the volume balance constraint. In the compressible case, it is often reasonable to linearize the volume balance constraint in time to develop a nonlinear parabolic pressure equation. Although this separation of the governing equations is not supported by any rigorous mathematical analysis, the splitting is heuristically motivated by model problems and computational experience.

Conventional numerical treatment of practical multi-dimensional, multi-phase fluid flow problems in enhanced oil recovery and aquifer remediation is very computationally expensive and may provide inadequate results. The primary impediments to simulation accuracy are standard numerical methods that cannot properly treat complicated physical and chemical flow mechanisms, and the employment of computational meshes that do not allow sufficient resolution of important flow features. To provide useful information for the development of recovery processes, field-scale simulations must be able to resolve complicated, fine-scale, localized (transient and static) flow behavior. Consequently, the computational mesh employed in a simulation must be sufficiently fine to resolve the length scales of important features. However, the systems of discrete equations resulting from the approximation of typical reservoir and fluid models

severely restricts the allowable refinement of the mesh. This is due to the magnitude of the discrete systems and the complicated nonlinear relationships amongst the variables. By combining AMR with high resolution numerical techniques, we can concentrate numerical effort near localized features. As a result, greater local resolution can be achieved more efficiently than if a globally fine mesh is used.

The adaptive local mesh refinement paradigm that we employ is based on the ideas introduced by Berger and Oliger, and extended by Berger, Colella and Trangenstein in separate pursuits. The AMR process automatically and dynamically generates a hierarchy of nested levels of computational cells. Within each level, the cells are maintained as a list of logically-rectangular patches. The advantages of this patch refinement approach over other refinement strategies are significant for a wide range of computational problems including, but not limited to, flow in porous media. The patch approach is designed so that a small number of computationally rich tasks can be organized in a highly structured fashion. We are able to use integration and iteration routines developed for rectangular meshes on all patches on all levels in the adaptive mesh hierarchy. Therefore, the user interface in the code appears much like it does in a conventional simulator. Moreover, the implementation of efficient and accurate high resolution numerical routines is better understood on logically-rectangular meshes. In addition, the numerical integration routines used to solve the equations on the adaptive mesh are easily separated from the structure of the AMR algorithm. This greatly enhances code extendibility, maintenance and generality. Finally, high resolution numerical methods and domain decomposition methods can be made efficient on vector and parallel computers when the data is organized in a logically-rectangular fashion as allowed in the patch refinement approach.

Our use of multilevel iteration to solve the pressure equation associated with flow in porous media is motivated differently than conventional applications of multigrid-type methods. We are primarily concerned with capturing fluid interfaces whose time evolution is modeled by the coupling between hyperbolic mass conservation equations and an elliptic/parabolic pressure equation. The placement of mesh refinement is governed by the need to resolve fluid interfaces and the complicated wave behavior presented by the hyperbolic system. The role of the multilevel iteration is to solve the pressure equation on the given adaptive mesh configuration. The iteration must be able to treat fairly general mesh configurations as well as reasonably general behavior in the system of discrete equations representing the pressure equation. In addition, physically-motivated up-scaling and mesh communication concerns need to be honored during the iteration process to ensure consistency of the pressure solution with that of the mass equations.

We will discuss algorithmic performance and various computational examples. In a typical two-dimensional problem, the overhead cost associated with AMR for inter-patch communication and mesh adaptivity is roughly 10% to 20% of the cost of the entire computation. As a result, our AMR code can complete a simulation in a fraction the time required for a comparable simulation in which the mesh is fine everywhere. For the incompressible case, we will present a two-phase polymer flooding model consisting of a system of nonlinear hyperbolic mass conservation equations coupled to an elliptic pressure equation. The presentation of the compressible case will center on a fully compositional reservoir model that describes local phase equilibrium through the minimization of the Gibbs free energy of the fluid mixture. We will also address issues concerning the development of a general object-oriented AMR methodology that exploits the advantages of C++ for the AMR program structure and data management.

Semi-Coarsening Multigrid Methods for Parallel Computing

Jim E. Jones
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center

Standard multigrid methods are not well suited for problems with anisotropic coefficients which can occur, for example, on grids that are stretched to resolve a boundary layer. There are several different modifications of the standard multigrid algorithm that yield efficient methods for anisotropic problems. In the paper, we investigate the parallel performance of these multigrid algorithms.

Multigrid algorithms which work well for anisotropic problems are based on line relaxation and/or semi-coarsening. In semi-coarsening multigrid algorithms a grid is coarsened in only one of the coordinate directions unlike standard or full-coarsening multigrid algorithms where a grid is coarsened in each of the coordinate directions. When both semi-coarsening and line relaxation are used, the resulting multigrid algorithm is robust and automatic in that it requires no knowledge of the nature of the anisotropy. This is the basic multigrid algorithm whose parallel performance we investigate in the paper. The algorithm is currently being implemented on an IBM SP2 and its performance is being analyzed. In addition to looking at the parallel performance of the basic semi-coarsening algorithm, we present algorithmic modifications with potentially better parallel efficency. One modification reduces the amount of computational work done in relaxation at the expense of using multiple coarse grids. This modification is also being implemented with the aim of comparing its performance to that of the basic semi-coarsening algorithm.

Petr Vanek,
Jan Mandel,
Marian Brezina
Center for Computational Mathematics
University of Colorado at Denver
Denver CO 80217-3364

An algebraic multigrid algorithm for symmetric, positive definite linear systems is developed based on the concept of prolongation by smoothed aggregation. Coarse levels are generated automatically. We present a set of requirements motivated heuristically by a convergence theory. The algorithm then attempts to satisfy the requirements. Input to the method are the coefficient matrix and zero energy modes, which are determined from nodal coordinates and knowledge of the differential equation. Efficiency of the resulting algorithm is demonstrated by computational results on real world problems from solid elasticity, plate bending, and shells.

Topic: *Applications*	*Session Chair:* *TBA*	*Room B*
10:30 - 11:00	A. Frommer	Lattice QCD Computations: Recent Progress with Modern Krylov Subspace Methods
11:00 - 11:30	R. Karamikhova	Numerical Solution of High-Kappa Model of Superconductivity
11:30 - 12:00	M. Heroux	The Impact of Improved Sparse Linear Solvers on Industrial Engineering Applications

Lattice QCD computations: Recent progress with modern Krylov subspace methods

Andreas Frommer
Fachbereich Mathematik
Bergische Universität GH Wuppertal
42097 Wuppertal, Germany

Abstract

Quantum chromodynamics (QCD) is the fundamental theory of the strong interaction of matter. In order to compare the theory with results from experimental physics, the theory has to be reformulated as a discrete problem of *lattice gauge theory* using stochastic simulations. The computational challenge consists in solving several hundreds of very large linear systems with several right hand sides. A considerable part of the world's supercomputer time is spent in such QCD calculations.

This talk considers solving systems for the *Wilson fermions*. We review some recent progress on the algorithmic level obtained in our cooperation with partners form theoretical physics. The Wilson fermion matrix M is of the form

$$M = I - \kappa D,$$

where D is a matrix with stochastic complex entries resulting from a nearest neighbour coupling between grid points on a 4-dimensional time-space lattice. Each grid point holds 12 unknowns. Typical grid sizes are 16^4 to 32^4 so that M has dimension up to $12 \cdot 10^6$. The hopping parameter $\kappa \in \mathbf{R}$ varies in an interval $[0, \kappa_c)$, where κ_c is the first value for which M becomes singular. The matrix M is non-hermitian, but exposes a so-called γ_5-*symmetry*

$$\gamma_5 M^H = M \gamma_5$$

with a simple hermitian matrix γ_5 representing a permutation of variables within each grid point.

The γ_5-symmetry of M and its particular dependence on κ can both be exploited to simplify the Lanczos process used in generating a basis of the Krylov subspaces associated with the initial residual. We will show how this

yields more efficient solvers for the Wilson fermion matrix as compared to the popular standard methods in QCD.

Moreover, we will address the issue of parallel preconditioning. In particular, we will present a new ordering for the SSOR preconditioner which yields significant improvements upon standard red-black SSOR. This preconditioner exposes a medium grain parallelism which can be exploited on MIMD machines as well as on some special purpose machines like the *Quadrics* SIMD computers used in QCD.

Numerical Solution of High-kappa Model of Superconductivity[2]

Rossitza Karamikhova[1]

Abstract

We present formulation and finite element approximations of High-kappa model of superconductivity which is valid in the high κ, high magnetic field setting and accounts for applied magnetic field and current. Major part of this work deals with steady-state and dynamic computational experiments which illustrate our theoretical results numerically. In our experiments we use Galerkin discretization in space along with Backward-Euler and Crank-Nicolson schemes in time. We show that for moderate values of κ, steady states of the model system, computed using the High-kappa model, are virtually identical with results computed using the full Ginzburg-Landau (G-L) equations. We illustrate numerically optimal rates of convergence in space and time for the $\mathcal{L}^2$ and $\mathcal{H}^1$ norms of the error in the High-kappa solution. Finally, our numerical approximations demonstrate some well-known experimentally observed properties of high-temperature superconductors, such as appearance of vortices, effects of increasing the applied magnetic field and the sample size, and the effect of applied constant current.

1 Introduction

Throughout this work $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ denotes a bounded region occupied by the superconducting sample, Γ is the Lipschitz continuous boundary of Ω, and $\mathbf{n}$ is the unit outer normal vector to Γ. The High-kappa model presented in [7], see also [2], and [3], consist of the following two leading order systems for the vector-valued magnetic potential $\mathbf{A}$, and the complex-valued order parameter ψ, respectively:

$$\begin{aligned} \mathbf{curl}\ \mathrm{curl}\ \mathbf{A} &= (0, J)^T \ \text{ in } \ \Omega, \\ \mathbf{A}\cdot\mathbf{n} &= 0 \ \text{ on } \ \Gamma, \\ \mathrm{curl}\ \mathbf{A} &= H - Jx \ \text{ on } \ \Gamma, \end{aligned} \tag{1}$$

$$\begin{aligned} \psi_t - \left(\frac{\xi}{l}\right)^2 \Delta\psi + |\mathbf{A}|^2\psi + (i\Phi - 1)\psi \ &+ \ |\psi|^2\psi + 2i\left(\frac{\xi}{l}\right)\mathbf{A}\cdot\nabla\psi = 0 \quad \text{on} \quad \Omega\times[0,\infty), \\ \nabla\psi\cdot\mathbf{n} &= 0 \quad \text{on} \quad \Gamma\times[0,\infty), \\ \psi(0) &= \psi^0 \quad \text{in} \quad \Omega, \end{aligned} \tag{2}$$

where λ and ξ are two temperature dependent microscopic characteristic lengths called penetration depth and coherence length, respectively, l is length scale used in nondimensionalization of the equations, σ is a microscopic parameter, $\Phi = -\left(\frac{l^2 J}{\sigma\xi\lambda}\right) y$ is the scalar electric potential, $\mathbf{H} = (0, 0, H)^T$ is the applied magnetic field, and $\mathbf{J} = (0, J, 0)^T$ is the applied constant current.

In what follows we will use the spaces $\mathbf{H^m}(\Omega) = [H^m(\Omega)]^d$, and $\mathbf{H^1_n}(\Omega) = \{\mathbf{Q} \in \mathbf{H^1}(\Omega) \mid \mathbf{Q}\cdot\mathbf{n} = 0 \ \text{ on } \ \Gamma\}$. Then, solution $\mathbf{A}$ of the leading order system (1) satisfies the following weak formulation:

$$\text{Seek } \mathbf{A} \in \mathbf{H^1_n}(\Omega) \text{ such that } \quad \bar{a}(\mathbf{A}, \tilde{\mathbf{A}}) = F(\tilde{\mathbf{A}}) \ \ \forall\, \tilde{\mathbf{A}} \in \mathbf{H^1_n}(\Omega), \tag{3}$$

[1]Department of Mathematics, University of Texas at Arlington, Box 19408, Arlington, TX 76019-0408. E-mail: rossi@utamat.uta.edu. This work is based on the Ph.D. thesis of the author completed in August 1995 at Virginia Tech.

[2]This work was supported by the Department of Energy through contract DE-FG0593ER25175.

where $\bar{a}(\cdot,\cdot)$ is the bilinear form

$$\bar{a}(\mathbf{A},\tilde{\mathbf{A}}) = (\text{curl } \mathbf{A}, \text{curl } \tilde{\mathbf{A}}) + (\text{div } \mathbf{A}, \text{div } \tilde{\mathbf{A}}),$$

and $F(\cdot)$ is the linear functional

$$F(\tilde{\mathbf{A}}) = (H - Jx, \text{curl } \tilde{\mathbf{A}}.$$

Denote $\mathcal{V} = L^{\infty}\left(0,T;\mathcal{H}^1(\Omega)\right) \cap \mathcal{H}^{1=}(0,T;L^2(\Omega))$. Then, solution ψ of the time-dependent leading order system (2) satisfies the following weak formulation:

Seek $\psi \in \mathcal{V}$ such that

$$\begin{aligned} (\psi_t, v) + a(\psi, v) - b(\mathbf{A}, v, \psi) &= (f(\psi), v) \quad \text{for all} \quad v \in \mathcal{H}^1(\Omega), t \geq 0, \\ \psi(0) &= \psi^0, \end{aligned} \tag{4}$$

where $a(\cdot,\cdot)$ is the bilinear form

$$a(\psi, v) = \left(\frac{\xi}{l}\right)^2 (\nabla\psi, \nabla v) + (\psi, v),$$

$b(\cdot,\cdot,\cdot)$ is the trilinear form

$$b(\mathbf{A}, \psi, v) = 2i\left(\frac{\xi}{l}\right)(\mathbf{A}\cdot\nabla\psi, v),$$

and $f(\cdot)$ is a nonlinear function of the form

$$f(\psi) = \left(2 - i\Phi - |\psi|^2 - |\mathbf{A}|^2\right)\psi.$$

Given a uniformly regular triangulation $\mathcal{T}_h$ of $\bar{\Omega}$ and a fixed integer $r \geq 1$, we consider standard finite element spaces $S_r^h = \{v_h \in C^0(\bar{\Omega});\ v_h|_{\mathcal{K}} \in P_r \ \ \forall\, \mathcal{K} \in \mathcal{T}_h\}$. Then we set $V^h = \mathcal{S}_r^h$, and $\mathbf{W}^h = \mathbf{S}_r^h \cap \mathbf{H}_n^1(\Omega)$, where $\mathcal{S}_r^h$ and $\mathbf{S}_r^h$ denote the corresponding complex and vector spaces, respectively. For the purpose of our analysis, see [1], and [7], assume that:

$$\begin{aligned} \lim_{h\to 0}\left(\sup_{t\in[0,T]}\inf_{v_h\in V^h}\left(\|\psi(t) - v_h\|_{\infty} + h^{-\frac{d}{2}}\|\psi(t) - v_h\|_0\right)\right) &= 0, \quad \text{and} \\ \lim_{h\to 0}\left(\inf_{\mathbf{w}_h\in\mathbf{W}^h}\left(\|\mathbf{A} - \mathbf{w}_h\|_{\infty} + h^{-\frac{d}{2}}\|\mathbf{A} - \mathbf{w}_h\|_0\right)\right) &= 0. \end{aligned}$$

The approximate problem for $\mathbf{A}_h$ reads:

$$\text{Seek } \mathbf{A}_h \in \mathbf{W}^h \text{ such that } \quad \bar{a}(\mathbf{A}_h, \tilde{\mathbf{A}}_{\mathbf{h}}) = F(\tilde{\mathbf{A}}_{\mathbf{h}}) \ \ \forall\, \tilde{\mathbf{A}}_{\mathbf{h}} \in \mathbf{W}^h. \tag{5}$$

Let $k > 0$ denote a fixed time step, $t_n = nk$, for $0 \leq n \leq N$, and let $T = Nk$. In what follows we will use Ψ_1^n to denote the fully discrete Backward-Euler-Galerkin (B-E-G) approximation of $\psi(t_n)$ at time t_n. Similarly, Ψ_2^n will denote the Crank-Nicolson-Galerkin (C-N-G) approximation of $\psi(t_n)$. These approximations satisfy the following two discrete problems:

B-E-G: Seek $\Psi_1^n \in V^h$ such that

$$\begin{aligned} (\frac{\tilde{d}\Psi_1^n}{dt}, v_h) + a(\Psi_1^n, v_h) - b(\mathbf{A}_h, v_h, \Psi_1^n) &= (f_h(\Psi_1^n), v_h) \ \ \forall\, v_h \in V^h\ ,\ \ 1 \leq n \leq N, \\ \Psi_1^0 &= \tilde{\psi}^0. \end{aligned} \tag{6}$$

C-N-G: Seek $\Psi_2^n \in V^h$ such that

$$(\frac{\tilde{d}\Psi_2^n}{dt}, v_h) + a(\bar{\Psi}_2^n, v_h) - b(\mathbf{A}_h, \bar{\Psi}_2^n, v_h) = (f_h(\bar{\Psi}_2^n), v_h) \quad \forall\, v_h \in V^h,\ 1 \le n \le N,$$
$$\Psi_2^0 = \tilde{\psi}^0, \tag{7}$$

where we denote $\frac{\tilde{d}}{dt}\Psi_s^n = \left(\Psi_s^n - \Psi_s^{n-1}\right)/k$ for $s = 1,\ 2$, and $\bar{\Psi}_2^n = \left(\Psi_2^n + \Psi_2^{n-1}\right)/2$.

Existence, uniqueness, and error estimates for problem (5) have been established in [7]. In particular, for $\mathbf{A} \in \mathbf{H}_n^1(\Omega) \cap \mathbf{H}^{m+1}(\Omega)$ we have the following optimal $\mathbf{H}^1$ and $\mathbf{L}^2$ estimates for the error $\mathbf{E}_h = \mathbf{A} - \mathbf{A}_h$:

$$\|\mathbf{E}_h\|_j \le Ch^{m+1-j}\|\mathbf{A}\|_{m+1},\ ;\ j = 0, 1.$$

Errors in (6) and (7) are defined as $e_s^n = \Psi_s^n - \psi(t_n)$, where $s = 1,\ 2$, respectively. The following theorem establishes optimal error estimates for the approximations Ψ_1^n and Ψ_2^n of $\psi(t_n)$. The proof of this theorem can be found in [7], and is based on some results of [1] and [8].

Theorem 1.1 *Let ψ denote the solution of (4), and for $1 \le n \le N$ let Ψ_s^n, $s = 1,\ 2$ satisfy (6) or (7),respectively. For a positive integer m with $m \ge d/2$ we assume that $\psi \in \mathcal{V}$ with $\psi(\cdot, \mathbf{x}) \in \mathcal{H}^{m+1}(\Omega)$, $\|\psi_{ttt}\|_0 \le C$, and $\|\Delta\psi_{tt}\|_0 \le C$. If $\mathbf{A} \in \mathbf{H}_n^1(\Omega) \cap \mathbf{H}^{m+1}(\Omega)$ and if $\|e_s^0\|_j \le Ch^{m+1-j}$; $j = 0, 1$, then for sufficiently small h and for $k = \mathrm{o}\left(h^{\frac{d}{2s}}\right)$ there exists unique solution Ψ_s^n of (6) or (7), respectively. Moreover the following optimal estimates are valid*

$$\|e_s^n\|_j \le C(h^{m+1-j} + k^s)\ ;\ j = 0, 1, \tag{8}$$

where $s = 1,\ 2$ for the B-E-G and the C-N-G problems, respectively □.

2 Numerical Results

Numerical solution of the full G-L equations, especially for simulation of vortices, has been successful only in the recent years, see e.g. [4], [5], and [6]. The objective of our computational experiments is to study vortex dynamics in type II superconductors under constant applied field and current. This is accomplished using a two dimensional finite element code based on the High-kappa model. Since the leading order time independent linear system for $\mathbf{A}$ is decoupled from the leading order time-dependent nonlinear system for ψ, one may first solve (1) and then substitute the solution into (2). This valuable computational property of the High-kappa model simplifies significantly numerical experiments both in terms of speed and storage requirements.

We present numerical results based on simulations with square superconducting samples having sides equal to 10ξ, 20ξ and 30ξ, where $\xi = 0.1l$. We often refer to such samples as a 1x1, 2x2, 3x3 sample. Spatial discretization in problems (3) and (4) is by piecewise biquadratic elements on a uniform grid having a given number of grid lines in x and y directions. For example, a 13x13 mesh means that there are 13 grid lines in each coordinate direction. For time discretization of system (4) we use B-E-G and C-N-G schemes, see (6) and (7), respectively. Initial conditions in all experiments correspond to a perfect superconducting sample characterized by $|\psi| = 0$. Numerical results are visualized using contour plots of magnitude of the order parameter at a fixed moment in time. Figures representing time evolution of order parameter should be read from left to right and top to bottom.

The linear system for $\mathbf{A}_h$ has symmetric and positive definite banded coefficient matrix, and is solved using banded Cholesky factorization. The nonlinear systems for Ψ_s^n, $s = 1\ ,2$; are solved using Newton's method. As a result, at each iteration one has to solve a linear system of algebraic equations which is symmetric and positive definite, if there is no applied current. However, in the applied constant current case the linearized system has a positive definite, but nonsymmetric coefficient matrix. To solve this system we use banded LU decomposition.

2.1 Steady-State Numerical Results

In steady-state numerical simulations (no applied current), we apply a constant (in space and time) magnetic field $\mathbf{H}$ directed perpendicular to the cross-section of the superconducting sample. As a result, vortices

Table 1: Maximum and minimum values of the magnetic field for different κ and applied fields H_e.

κ	H_e/κ	H_{max}/κ	H_{min}/κ
3	0.3	0.2961	0.2868
5	0.3	0.3002	0.2943
20	0.5	0.5005	0.4906
30	0.3(3)	0.3337	0.3271

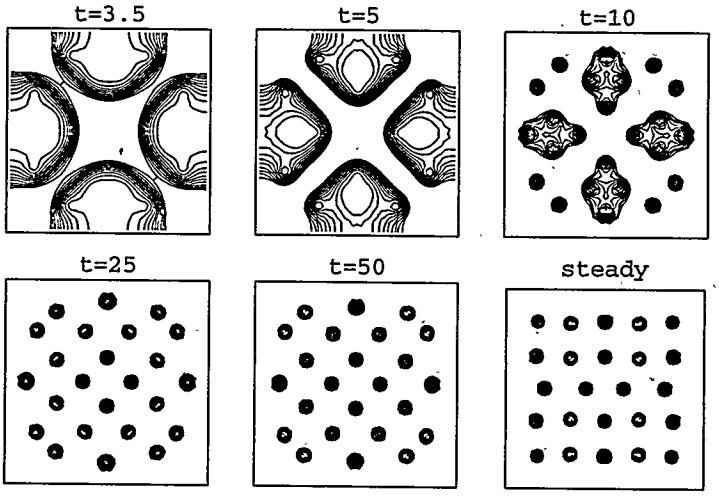


Figure 1: Time evolution for a 2x2 sample using the High-kappa model with $H_0 = 0.5$; B-E-G scheme.

appear and time evolution leads to steady-state, in which vortices form a hexagonal lattice pattern in order to minimize the Gibbs free energy of the system, see Figure 1 and Figure 4.

To demonstrate numerically that our model is valid for high values of the G-L parameter $\kappa = \lambda/\xi$, we fix the applied field $H = 0.5$ and generate steady-state solutions using increasing values of κ in the full G-L equations. These solutions are then compared with the steady-states solutions from the High-kappa model. In Figure 2 we present results for a 2x2 sample using $\kappa = 5$ and 100 in the full G-L model, and the steady-state solutions for our model valid for $\kappa = \infty$. It is evident that for values of κ exceeding 5 the contour plots appear virtually identical.

In Table 1 we examine the magnetic field in the superconductor for different values of κ and the applied field H_e. Maximum and minimum values of the magnetic field computed from full G-L equations at various values of κ demonstrate that, as expected, for large values of κ the magnetic field in the superconductor is nearly constant, and that it is equal to the applied field.

Next, we turn our attention to computations with increasing applied fields, which are known experimentally to cause appearance of more vortices. We demonstrate this phenomena numerically in Figure 3, where we present steady-state configurations for a 1x1 sample using $H = 0.3, 0.5, 0.7$, respectively. Another experimentally observed phenomena is the significant increase in the number of vortices resulting from increasing the sample size. This phenomena is demonstrated numerically in Figure 4, where we present steady-state configurations for 1x1, 2x2 and 3x3 samples using fixed $H = 0.3$.

A task of critical importance in computations is to avoid local minimizers of Gibbs free energy. Our experiments indicate that this issue is clearly related to the choice for the Newton's method tolerance Tol in computations. In Figure 4 presents results using $Tol = .5 \times 10^{-4}$ and $Tol = .5 \times 10^{-11}$. Each row represents steady-states obtained using a fixed Newton tolerance Tol, and each column corresponds to a fixed value $H = 0.3, 0.5$, and 0.7, respectively. Figure 5 demonstrates that, indeed, allowing for larger tolerances in the Newton iteration, the steady-state criteria may be satisfied at a local minimum of the system's energy.

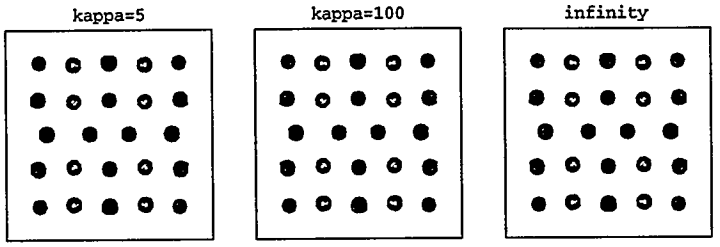


Figure 2: Steady-state vortex configurations for a 2x2 sample using fixed H=0.5 and increasing values of κ.

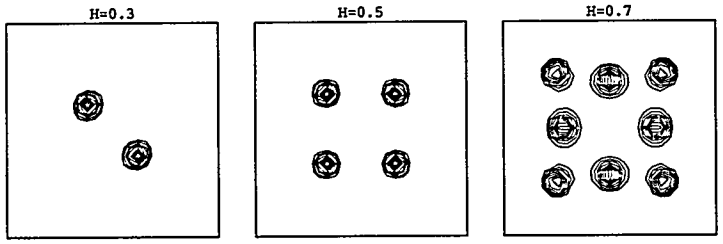


Figure 3: Steady-state configurations for a 1x1 sample using increasing values of the applied field H.

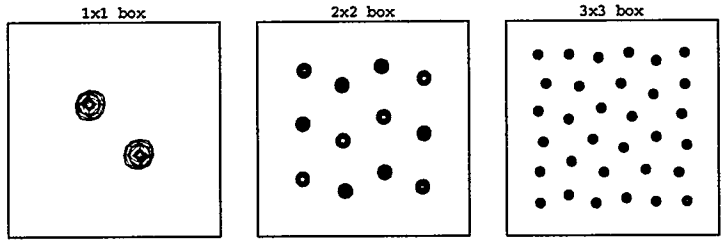


Figure 4: Steady-state configurations using fixed $H = 0.3$, and increasing the sample size.

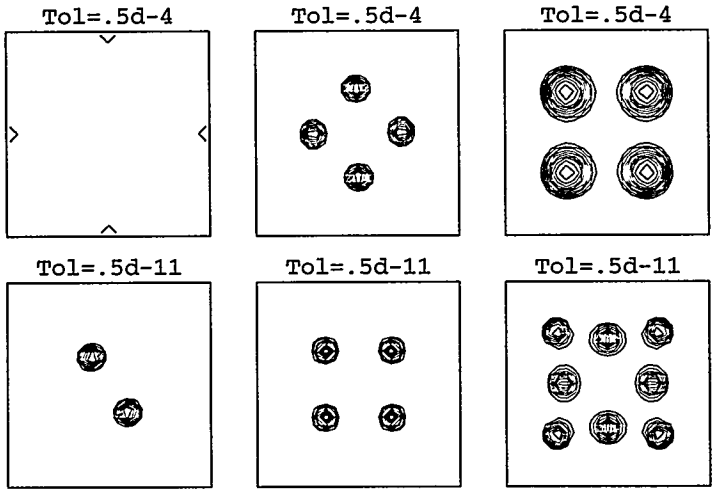


Figure 5: Steady-state configurations for a 1x1 sample using different values for Tol, and H=0.3, 0.5, 0.7.

Table 2: Spatial rates of convergence in the High-kappa model.

$\mathcal{L}^2$ rates in space			
Time	$R_{7,13}$	$R_{13,25}$	$R_{25,39}$
$3.5+10^{-7}$	3.22	3.16	3.03
$10.5+10^{-7}$	3.26	3.15	3.02
$\mathcal{H}^1$ error rates			
Time	$R_{7,13}$	$R_{13,25}$	$R_{25,39}$
$3.5+10^{-7}$	2.12	2.08	1.98
$10.5+10^{-7}$	2.15	2.08	1.98

Table 3: Temporal rates of convergence in the High-kappa model

$\mathcal{L}^2$ rates in space			
Time discretization	time=1.6	time=4.1	time=11.1
B-E	0.97	1.04	1.03
C-N	2.12	2.03	2.21
$\mathcal{H}^1$ error rates			
Time discretization	time=1.6	time=4.1	time=11.1
B-E	0.95	1.01	1.07
C-N	2.16	1.98	1.93

2.2 Numerical Approximations of Convergence Rates

We continue with numerical results concerning approximation of the $\mathcal{L}^2$ and $\mathcal{H}^1$ rates of convergence in both space and time. We show that the optimal rates established in Theorem 1.1, see (8), are also valid numerically. Our task here is complicated by the absence of a closed form exact solution for our model, which can be used to estimate the actual rates of convergence. To overcome this difficulty we compute "exact" solution using a "very" fine grid and appropriate time step.

In Table 2 we present numerical approximations of the spatial rates. We generate initial states using a 69x69 grid at times: $t = 3.5$ and $t = 10.5$. Then we make one "very small" time step $k = 10^{-7}$ to obtain our "exact" solution at times $t = 3.5 + k$ and $t = 10.5 + k$. Next we use the initial guesses at $t = 3.5$ and $t = 10.5$ as starting points on a sequence of coarser meshes having 7x7, 13x13, 25x25 and 39x39 grid lines. After computing the errors corresponding to the approximate solutions we calculate rates of convergence in the usual manner. Table 2 indicates a very good agreemen between the numerical approximations for the spatial rates and the optimal theoretical values of $O\left(h^2\right)$ in $\mathcal{H}^1$ and $O\left(h^3\right)$ in $\mathcal{L}^2$ using biquadratics.

Next we turn to numerical approximations of the $\mathcal{L}^2$ and $\mathcal{H}^1$ rates in time. We expect to observe $O(k)$ for B-E and $O(k^2)$ for C-N scheme. In Table 3 we present results obtained at three different moments in time: t= 1.6, t=4.1, and t=11.1. Corresponding initial states at times t=1.0, t=3.5, and t=10.5 were generated using a 39x39 grid, fixed time step $k = 0.5$, and prescribed number of $nt = 2, 7$ and 21 steps, respectively. An "exact" solution was obtained using 39x39 mesh, starting with the initial states, and using fixed $k = 0.01$ and $nt = 60$. Two approximate solutions are computed starting from each of the above initial states using $k = 0.1, 0.3$, and $nt = 6,\ 2$, respectively. Results presented in Table 3 are again in a very good agreement with the optimal theoretical values.

2.3 Vortex Dynamics Under Applied Constant Current

Numerical results presented in previous sections deal with the case when the superconducting sample is subjected only to an applied magnetic field. Below, we present results for the case, when in addition to the applied magnetic field there is applied constant current of magnitude J. It is experimentally established that if current is passed through a type II superconductor (in the presence of applied magnetic field), then vortices begin to move in a direction transverse to the transport current. As a result of the viscous flow of vortices, energy is dissipated. This induces voltage, produces resistance in the sample, and hence destroys

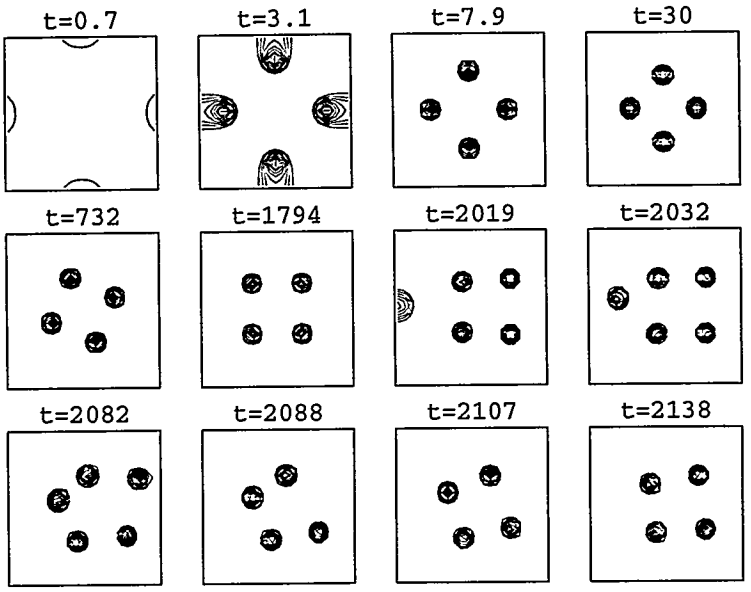


Figure 6: Vortex dynamics for a 1x1 sample using $H = 0.5$ and $J = 0.01$; current is applied after steady-state is reached.

superconductivity. Vortex movement can be clearly observed in Figure 6, which corresponds to a simulation for a 1x1 sample with $H = 0.5$ and $J = 0.01$. At the initial moment only magnetic field is applied. After steady-state vortex configuration has been formed current of magnitude J is applied in the direction of the y-axis. As a result vortices begin to move at right angles to the current flow (from left to right across the sample).

References

[1] G. Akrivis, V. Dougalis, and O. Karakashian, *On Fully Discrete Galerkin Methods of Second-order Temporal Accuracy for the Nonlinear Schrodinger Equation*, **Numerishe Mathematik**, 1991.

[2] Q. Du and P. Gray, *High-Kappa Limits of the Time-Dependent Ginzburg-Landau Model*, Preprint, 1995.

[3] S. J. Chapman, Q. Du, M. D. Gunzburger, and J. S. Peterson, *Simplified Ginzburg-Landau Models for Superconductivity Valid for High-kappa and High Fields*, to appear in **SIAM Journal of Applied Math.**

[4] M. Doria, J. Gubernatis, D. Rainer, *Solving the Ginzburg-Landau Equations by Simulated Annealing*, **Phys. Rev. B**, 41, pp. 6335-6340, 1990.

[5] Q. Du and M. Gunzburger, *A Model for Superconducting Thin Films Having Variable Thickness*, **Physica D**, 69, pp. 215-231, 1993.

[6] Q. Du, *Finite Element Methods for the Time-Dependent Ginzburg-Landau Model of Superconductivity*, **Computers Math. Applic.**, 27, No 12, pp. 119-133, June 1994.

[7] R. Karamikhova, **A Finite Element Analysis of a High-kappa, High Field Ginzburg-Landau Model of Superconductivity**, Ph.D. thesis, Virginia TECH, 1995.

[8] V. Thomée, **Galerkin Finite Element Methods for Parabolic Problems**, Springer Verlag, New York, 1984.

The Impact of Improved Sparse Linear Solvers on Industrial Engineering Applications

Mike Heroux
Leader, Scalable Computing and Algorithms Group
Applications Division
Cray Research, Inc.
655 Lone Oak Drive
Eagan, MN 55121
email: mike.heroux@cray.com

Majdi Baddourah
Eugene L. Poole
Chao Wu Yang
Applications Division, Cray Research, Inc.

There are usually many factors that ultimately determine the quality of computer simulation for engineering applications. Some of the most important are the quality of the analytical model and approximation scheme, the accuracy of the input data and the capability of the computing resources. However, in many engineering applications the characteristics of the sparse linear solver are the key factors in determining how complex a problem a given application code can solve. Therefore, the advent of a dramatically improved solver often brings with it dramatic improvements in our ability to do accurate and cost effective computer simulations.

In this presentation we discuss the current status of sparse iterative and direct solvers in several key industrial CFD and structures codes, and show the impact that recent advances in linear solvers have made on both our ability to perform challenging simulations and the cost of those simulations. We also present some of the current challenges we have and the constraints we face in trying to improve these solvers. Finally, we discuss future requirements for sparse linear solvers on high performance architectures and try to indicate the opportunities that exist if we can develop even more improvements in linear solver capabilities.

Topic: *Projection Method*	*Session Chair:* *Roland Freund*	*Room C*
10:30 - 11:00	A. Popov	Projection Preconditioning for Lanczos-Type Methods
11:00 - 11:30	R. Bramley	Partial Row Projection Methods
11:30 - 12:00	P. Kolm	Generalized Subspace Correction Methods

PROJECTION PRECONDITIONING FOR LANCZOS-TYPE METHODS

S.S.Bielawski, S.G.Mulyarchik,
Radio Physics and Electronics Faculty, Belarusian State University, 4 F.Scoriny Ave., Minsk, 220080, Belarus

and *A.V.Popov*
Radio Physics and Electronics Faculty, Belarusian State University, 4 F.Scoriny Ave., Minsk, 220080, Belarus
Present address: The Economics Institute, University of Colorado, 1005 12th. St., Boulder, CO, 80302, USA
e-mail: andrei@ucsub.colorado.edu

We show how auxiliary subspaces and related projectors may be used for preconditioning nonsymmetric system of linear equations. It is shown that preconditioned in such a way (or projected) system is better conditioned than original system (at least if the coefficient matrix of the system to be solved is symmetrizable). Two approaches for solving projected system are outlined. The first one implies straightforward computation of the projected matrix and consequent using some direct or iterative method. The second approach is the projection preconditioning of conjugate gradient-type solver. The latter approach is developed here in context with biconjugate gradient iteration and some related Lanczos-type algorithms. Some possible particular choices of auxiliary subspaces are discussed. It is shown that one of them is equivalent to using colorings. Some results of numerical experiments are reported.

1. Introduction. Among the conjugate gradient (CG)-type methods for solving nonsymmetric systems of linear equations $Ax = b$, biconjugate gradient (BCG) iteration [10] is one of the most popular algorithms. Despite the absence of minimization property, the BCG method and related algorithms CGS [18], CGSTAB [20] outperform the others in many scientific and engineering applications. Therefore, Lanczos-type methods are widely used and in a rapid further development. In particular, the QMR [6] and related algorithms are based on quasiminimization principle. Apart from more smooth convergence behavior, they overcome some kinds of breakdown of the underlying Lanczos process, especially when coupled with a look-ahead technique. Composite step approach [2] is another technique for improving the erratic convergence history of Lanczos-type methods as well as for avoiding a pivot breakdown. For more information of Lanczos-type methods and techniques for avoiding breakdown see [8], [3].

Here we describe yet another way for improving biconjugate gradients. It uses auxiliary subspaces and projection operators. Projected algorithms of the BCG and related methods are proposed. The approach under consideration may be regarded as preconditioning by means of projector. Such preconditioning may be applied by itself or in conjunction with one of commonly used preconditioners as considered in [7]. The present paper generalizes the technique developed in [15] to nonsymmetric linear systems. We will show that projected algorithm of Lanczos-type method may be more effective than standard one.

It should be pointed out that developed approach is closely related to deflation of conjugate gradients [16], where the projection preconditioner is built from linearly independent vectors. For practical application of this technique it is convenient to orthogonalize these vectors with respect to the inner product $u^T Av$ [11]. However, Gram-Schmidt procedure may be very expensive as the number of vectors under orthogonalization is large. If the system to be solved arises from discretization of partial differential equation (PDE), then deflation technique may be implemented effectively in context with domain decomposition method by exploiting some knowledge of the PDE, the mesh, and the differencing technique used [12], [13]. Although the lower bound of the spectrum may be increased, this approach is not sufficiently general since the above mentioned apriori knowledge are usually not available.

In contrast to deflation method, we will build the projectors from vectors which are already A-biorthogonal. Projected versions of BCG, CGS, and CGSTAB are investigated in conjunction

with some particular projection preconditioners are developed in this way using red-black and some linear orderings. Results of some numerical experiments are reported.

2. Projected System and Projection Preconditioning. Suppose we would like to solve the following two related linear systems

(2.1) $$Ax = b,$$

(2.2) $$A^T \hat{x} = \hat{b},$$

where A is general n by n nonsymmetric matrix and we are given initial guesses $x^* = 0$ and $\hat{x}^* = 0$ for convenience. If nonzero x and $\hat{x}^*$ are available then we could set b and $\hat{b}$ are equal to the corresponding residuals and use zero initial approximations.

The basic idea of the technique we develop is in exploiting the properly chosen auxiliary subspaces. To determine them we assume the vectors

(2.3) $$p_0, p_1, \ldots, p_{k-1}; \quad \hat{p}_0, \hat{p}_1, \ldots, \hat{p}_{k-1}$$

to be given, where $k<n$. We will assume throughout that (2.3) satisfy to the standard biconjugacy condition

$$\hat{p}_i^T A p_j = p_j^T A^T \hat{p}_i = 0, \quad i \neq j; \quad \hat{p}_i^T A p_i = p_i^T A^T \hat{p}_i \neq 0.$$

We denote the $n \times k$ matrices of which columns are the vectors (2.3) as P and $\hat{P}$ respectively. Denote also $E = span\{p_0, p_1, \ldots, p_{k-1}\}$, $F = span\{\hat{p}_0, \hat{p}_1, \ldots, \hat{p}_{k-1}\}$. Or, in still the other words, E is the column space of P and F is that of $\hat{P}$. Now $\Re^n$ may be represented as $\Re^n = E \oplus E*$ and $\Re^n = F \oplus F*$, where E^* is an orthogonal complement of the column space of $A^T \hat{P}$ and F^* is that of AP, see [19]. Then it is naturally to introduce two pairs of complementary projectors

(2.4a) $$Q = P\left(\hat{P}^T A P\right)^{-1} \hat{P}^T A,$$

(2.4b) $$R = I - Q$$

and

(2.4c) $$\hat{Q} = \hat{P}\left(P^T A^T \hat{P}\right)^{-1} P^T A^T,$$

(2.4d) $$\hat{R} = I - \hat{Q},$$

where I denotes the $n \times n$ identity matrix. Here Q (alternatively, R) is the projector onto E (alternatively, E^*) along E^* (alternatively, E). The other pair of projectors uses F and F^* likewise.

The following equalities may be easily verified by direct computations

(2.5a) $$AQ = \hat{Q}^T A = \hat{Q}^T A Q, \quad AR = \hat{R}^T A = \hat{R}^T A R,$$

(2.5b) $$A^T \hat{Q} = Q^T A^T = Q^T A^T \hat{Q}, \quad A^T \hat{R} = R^T A^T = R^T A^T \hat{R}.$$

The following statement we present here without proof.

Theorem 2.1. Let biconjugate vectors (2.3) be given. Then the solution of (2.1) may be expressed as

(2.6) $$x = y + z,$$

where $y \in E$ is given by

(2.7) $$y = P\left(\hat{P}^T A P\right)^{-1} \hat{P}^T b,$$

and $z \in E^*$ can be found from

(2.8) $$Az = \hat{R}^T b.$$

Furthermore, for the solution of the adjoint system (2.2) we have

(2.9) $$\hat{x} = \hat{y} + \hat{z},$$

where $\hat{y} \in F$ is given by

(2.10) $$\hat{y} = \hat{P}\left(P^T A^T \hat{P}\right)^{-1} P^T \hat{b},$$

and $\hat{z} \in F^*$ can be found from

(2.11) $$A^T \hat{z} = R^T \hat{b}.$$

The solution of (2.2) is not usually of interest. Therefore, we will focus on solving (2.1). Since $z = Rx$ then

(2.12) $$ARx = \hat{R}^T b$$

may be considered instead of (2.8). It will be referred throughout as projected system. In (2.12) AR is singular. However, this system is compatible because of (2.5). $rankAR = rank[AR, \hat{R}^T b] = rankR = n - k$. Clearly, (2.12) may be obtained from (2.1) by means of preconditioning from the left with the matrix $\hat{R}^T$.

Solution of (2.12) is not unique because of singularity of AR. Therefore, to obtain the solution of (2.1) by solving (2.12), we have to determine an additional condition. Theorem 2.1 gives it as (2.7).

Thus, the projection preconditioning implies three stages: (i) choose vectors (2.3) as a basis of an auxiliary subspace; (ii) compute the component $y \in E$ from (2.7); (iii) solve (2.12).

The developed technique is of no practical value unless the vectors (2.3) can be determined. We discuss the choice of these vectors in Section 4.

Let $\rho(X)$ be the spectral radius of a matrix X. The following rather obvious result deals with the conditionality of projected system regardless the particular choice of (2.3).

Lemma 2.2. If A is symmetrizable by similarity transformation

(2.13) $$\tilde{A} = T^{-1} A T,$$

where $\tilde{A}$ is symmetric and T is nonsingular, then

$$\rho(AR) \le \rho(A).$$

Thus, one may hope that iterative methods will converge more rapidly when applied to (2.12) than when applied to (2.1).

3. Projected Algorithm of Biconjugate Gradients. It is known that preconditioning makes the use of CG-type methods especially attractive in practice, see [7]. We restrict our attention by the only left preconditioning with the matrix C since right-hand-side preconditioning may be easily taken into account by means of appropriate change of the unknowns [1].

We now give the result related to preconditioning of A regardless particular choice of (2.3). In this case both standard and projected BCG algorithms need the same amount of arithmetics, associated with preconditioning.

Suppose matrix of the system to be solved can be symmetrized with a similarity transformation (2.13). We assume also that preconditioner may be given as

(3.1) $$C = (L + \tilde{D})\tilde{D}^{-1}(\tilde{D} + U),$$

where $L(U)$ is strictly lower (upper) triangular of A. A particular choice of diagonal matrix $\tilde{D}$ determines a special procedure from the class (3.1). The following statement is the extension of Lemma 2.2. We give it here without proof.

Theorem 3.1. Suppose A is symmetrizable by a similarity transformation (2.13). Suppose the preconditioner from the class (3.1) has a positive definite real part, i.e.

$$u^T H u > 0$$

for any $u \in \Re^n$, $H = 0.5\left(C + C^T\right)$. Then

$$\rho\left(C^{-1}AR\right) \le \rho\left(C^{-1}A\right).$$

We focus on the problem of solving (2.12). Obvious way is to compute the matrix AR and the vector $\hat{R}^T b$ explicitly. Then to solve projected system one can apply some direct or iterative technique. This approach is equivalent to straightforward computation of preconditioned system.

Another approach for solving (2.12) implies completing both sets of vectors (2.3). It may be provided by the BCG method. In this case it is sufficient to carry out at most $n-k$ iterations to meet the solution of (2.1) if exact arithmetics is assumed. The standard BCG method is changed meanwhile since every pair of direction vectors, generated by the BCG algorithm, must be A-biorthogonalized against corresponding sets of given vectors (2.3). This means appearance of two additional matrix-vector products with R and $\hat{R}$ inside the iteration loop. Furthermore, (2.7) and (2.10) should be used as initial guesses for initialization of both residuals. With these in mind we can reformulate the BCG method as follows.

Algorithm 3.1 (preconditioned projected BCG algorithm)

1. Construct the preconditioner C.
2. Set $r^* = b - Ax^*, \hat{r}^*$ is arbitrary, $(\hat{r}^*)^T r^* \neq 0$, e.g. $\hat{r}^* = r^*$ (or $\hat{r}^* = \hat{b} - A^T\hat{x}^*$), $p_0 = \hat{p}_0 = 0$, $\rho_0 = 1$.
3. Compute y and $\hat{y}$ from (2.7) and (2.10) respectively.
4. Set $x_0 = x^* + y, r_0 = r^* - Ay, \hat{r}_0 = \hat{r}^* - A^T\hat{y}$.
5. Iterate for $i = 1,2,\ldots$ until converge

5.1. Solve $C\tilde{t} = r_{i-1}$; 5.2. $t = R\tilde{t}$; 5.3. $\rho_i = \hat{r}_{i-1}^T t$; 5.4. $\beta_i = \rho_i / \rho_{i-1}$; 5.5. $p_i = t + \beta_i p_{i-1}$;

5.6. Solve $C^T\tilde{t} = \hat{r}_{i-1}$; 5.7. $t = \hat{R}\tilde{t}$; 5.8. $\hat{p}_i = t + \beta_i \hat{p}_{i-1}$; 5.9. $s = Ap_i$; 5.10. $\delta = \hat{p}_i^T s$;

5.11. $\alpha_i = \rho_i / \delta$; 5.12. $x_i = x_{i-1} + \alpha_i p_i$; 5.13. $r_i = r_{i-1} - \alpha_i s$; 5.14. $\hat{r}_i = \hat{r}_{i-1} - \alpha_i A^T \hat{p}_i$.

Here nonzero initial approximations are assumed for generality. Comparing Algorithm 3.1 to the standard BCG algorithm, it can be easily seen that the former will be more effective than the latter if total cost of new operations is covered by either (i) more rapid convergence because of Lemma 2.2; or (ii) reducing the total cost of iteration if vectors (2.3) are chosen properly.

4. Choice of Auxiliary Subspaces. General approach, described above, needs specific vectors (2.3). We will assume for the remainder of the paper that (2.1) arises from finite element or finite difference discretization of the second order elliptic PDE.

One possible goal in choosing vectors (2.3) is to improve the spectrum. This can be provided by exploiting some apriori knowledge (as mentioned above, see [12]) which is usually not available, however.

Another aim is to reduce the amount of arithmetics per iteration. We will use the idea of [11] which implies A-orthogonalization of some set of unit vectors. The question is how to choose these unit basis vectors? In [11] some knowledge of PDE under consideration were exploited. In contrast to [11], we will determine unit vectors according to graph coloring.

Suppose all the nodes, determining unit basis vectors, are marked by one color and numbered first. Then all the remaining nodes may be marked by another color and numbered. Under these circumstances (2.1) takes the form

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x' \\ x'' \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{4.1}$$

After application of two-sided Gram-Schmidt process we have $P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$, $\hat{P} = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \end{bmatrix}$, where

$P_1 \in \Re^{k\times k}, \hat{P}_1 \in \Re^{k\times k}, P_2 \in \Re^{(n-k)\times k}, \hat{P}_2 \in \Re^{(n-k)\times k}$. $P_1, \hat{P}_1$ are nonsingular upper triangulars, and $P_2 = \hat{P}_2 = 0$. Then (2.4b) and (2.4d) transform into

$$R = \begin{bmatrix} 0 & -A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}, \qquad \hat{R} = \begin{bmatrix} 0 & -A_{11}^{-T}A_{21}^{T} \\ 0 & I \end{bmatrix} \tag{4.2}$$

respectively. Furthermore, (2.7) and (2.10) may be represented as

$$y = \begin{bmatrix} A_{11}^{-1}b_1 \\ 0 \end{bmatrix}, \qquad \hat{y} = \begin{bmatrix} A_{11}^{-T}\hat{b}_1 \\ 0 \end{bmatrix} \tag{4.3}$$

respectively. Thus, to obtain y and $\hat{y}$ we should solve two systems of order k. Note, that projected system (2.12) now looks like

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 0 & -A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} x' \\ x'' \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}, \tag{4.4}$$

where $f = b_2 - A_{21}A_{11}^{-1}b_1$. Let us briefly consider how it can be solved.

The natural approach is to compute AR explicitly. Then instead of (4.4) we have to solve $\begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} x' \\ x'' \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}$, where S is Schur complement of A_{11}. This system is compatible, as it has already been mentioned above, and we can find $x'' \in \Re^{n-k}$ by solving

$$Sx'' = f. \tag{4.5}$$

After x'' has been obtained we have to compute the component z of the solution (2.1) by projecting $z = Rx = \begin{bmatrix} -A_{11}^{-1}A_{12}x'' \\ x'' \end{bmatrix}$, i.e. x' is of no practical value. Then from (2.6) and (4.3) it is clear that

$$x = \begin{bmatrix} A_{11}^{-1}b_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -A_{11}^{-1}A_{12}x'' \\ x'' \end{bmatrix}. \tag{4.6}$$

From (4.5) and (4.6) it can be easily seen that straightforward computation of AR leads to well-known reduced system approach which is usually used together with red-black [9] and generalized red-black [4] orderings. This is not surprising since in this case A_{11} is diagonal and Schur complement of A_{11} may be computed easily.

From Lemma 2.2 now follows that $\rho(S) \le \rho(A)$ which is in agreement with the result of [17] for symmetric positive definite case.

Next turn our attention to another way for solving (4.4). Let us consider, for instance, projected preconditioned BCG. Clearly $r_0 = \begin{bmatrix} 0 \\ r''_0 \end{bmatrix}$, $\hat{r}_0 = \begin{bmatrix} 0 \\ \hat{r}''_0 \end{bmatrix}$, where $r''_0 \in \Re^{n-k}, \hat{r}''_0 \in \Re^{n-k}$. Further, if C is preconditioner for A then $\tilde{t} = C^{-1}r_0 = \begin{bmatrix} \tilde{t}' \\ \tilde{t}'' \end{bmatrix}$, i.e., nonzero $\tilde{t}'$ is appeared. However, $\tilde{t}'$ is never used in further computations since $t = R\tilde{t} = \begin{bmatrix} -A_{11}^{-1}A_{12}\tilde{t}'' \\ \tilde{t}'' \end{bmatrix}$. It can be easily seen that $s = Ap_i = \begin{bmatrix} 0 & 0 \\ A_{21} & A_{22} \end{bmatrix} p_i$, i.e., s contains first k zero components. Then r_i is of the same nonzero structure as r_0. Similar relations may be obtained for $\hat{r}_i, A^T\hat{p}_i$. Obviously, both inner products are standard operations in $\Re^{n-k}$.

Thus, we have a significant decreasing amount of arithmetics per iteration if only solution of systems with A_{11} and A_{11}^T does not require substantial efforts. In contrast to the reduced system approach, this technique may be applied in conjunction with various colorings since A_{11} is never computed explicitly.

If we consider 2-linear (zebra) ordering and assume C to be an approximate Schur complement preconditioner then described technique reduces to that considered in [5]. This is because CG-type method in fact solves (4.5) rather than (2.1). Our experiments with zebra ordering also show that preconditioning of S yields more effective solver than preconditioning of A. In this case the order of Schur complement is approximately a half of the order of A as well as for red-black ordering. However, we can use another orderings to reduce the order of S even more. Then the problem of construction of the approximate Schur complement preconditioner faces some difficulties. In this case preconditioning of A might become efficient if the order of resulting Schur complement is small enough.

5. Numerical Experiments. In this section we consider a linear system as it arises from the discretization of non-self-adjoint elliptic PDE over a two-dimensional rectangular domain using standard five-point computational molecula.

ILU(0) factorization [14] of A is used throughout our experiments. The BCG, CGS, and CGSTAB methods are investigated. For every of them the following orderings are applied to determine auxiliary vectors (2.3): (i) natural ordering (i.e., unprojected algorithm); (ii) red-black ordering; (iii) 2-linear (zebra) ordering; (iiii) 3-linear ordering where two of the colors determine unit basis vectors.

Test problem was taken from semiconductor device simulation because of ill-conditionality of discrete current continuity equations. This is a submicron bipolar transistor, see [15] for more details. All the investigated algorithms are applied to solve both electrons and holes continuity equations. Grid size is 4818 points. Iterations were terminated as soon as $\|r_N\|_\infty / \|r_0\|_\infty < TOL$, where TOL is a given value of stopping tolerance and N is total number of iterations performed. All the experiments were carried out on an IBM PC AT 486 using double precision.

Tables I-III present N and normalized run time t for all the three investigated methods. It can be seen from these tables that projected methods are more effective than traditional versions. Moreover, an algorithm based on 3-linear ordering is comparable to that based on zebra ordering and approximate Schur complement preconditioning from the point of view of overall performance.

Table I

Total number of iterations and normalized run time for the projected BCG

	TOL=1.e-6				*TOL=1.e-12*			
Ordering	*electrons*		*holes*		*electrons*		*holes*	
	N	*t*	*N*	*t*	*N*	*t*	*N*	*t*
natural	107	1.00	64	1.00	129	1.00	105	1.00
red-black	101	0.78	48	0.62	127	0.81	101	0.79
2-linear	80	0.62	51	0.67	115	0.74	92	0.73
3-linear	70	0.53	48	0.61	92	0.58	77	0.59

6. Concluding Remarks and Topics of Further Study. We have shown how projected versions of Lanczos-type methods may be developed. In other words, projection preconditioning has been extended to nonsymmetric systems of linear equations. Some advantages of this approach have been shown. One possible way of constructing auxiliary subspaces has been investigated. The results of numerical experiments illustrate its effectiveness. However, there are another goals, which may be taken into account when auxiliary vectors are determining. One can try to reduce the

spectral condition number of the projected system (at least for the case when the matrix of the system to be solved is symmetrizable) by checking diagonal domination and choosing the rows (or columns) of A as a basis vectors with consequent orthogonalization. Furthermore, unit vectors may be determined during this procedure. Perhaps Gershgorin's theorem may also be helpful in this way.

Table II

Total number of iterations and normalized run time for the projected CGS

	TOL=1.e-6				*TOL=1.e-12*			
Ordering	*electrons*		*holes*		*electrons*		*holes*	
	N	*t*	*N*	*t*	*N*	*t*	*N*	*t*
natural	50	1.00	58	1.00	77	1.00	81	1.00
red-black	50	0.84	64	0.92	75	0.81	79	0.82
2-linear	42	0.71	65	0.94	71	0.78	75	0.78
3-linear	40	0.66	48	0.68	60	0.64	69	0.69

Table III

Total number of iterations and normalized run time for the projected CGSTAB

	TOL=1.e-6				*TOL=1.e-12*			
Ordering	*electrons*		*holes*		*electrons*		*holes*	
	N	*t*	*N*	*t*	*N*	*t*	*N*	*t*
natural	42	1.00	23	1.00	73	1.00	66	1.00
red-black	36	0.69	25	0.87	67	0.74	71	0.87
2-linear	35	0.68	26	0.92	62	0.69	54	0.66
3-linear	38	0.71	26	0.88	59	0.63	48	0.56

REFERENCES

[1] S.F.Ashby, T.A.Manteuffel, and P.E.Saylor, *SIAM J.Numer.Anal.*, 27 (1990) 1542-1568

[2] R.E.Bank and T.F.Chan, *Numer.Math.*, 66 (1993) 295-319

[3] C.Brezinski and H.Sadok, *Appl.Numer.Math.*, 11 (1993) 443-473

[4] E.F.D'Azevedo, P.A.Forsyth, and W.-P.Tang, *SIAM J.Matrix Anal.Appl.*, 13 (1992) 944-961

[5] H.C.Elman, *SIAM J.Sci.Statist.Comput.*, 10 (1989) 581-605

[6] R.W.Freund and N.M.Nachtigal, *Numer.Math.*, 60 (1991) 315-339

[7] G.H.Golub and C.F.van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989

[8] M.H.Gutknecht, *SIAM J.Matrix Anal.Appl.*, 13 (1992) 594-639, 15 (1994) 15-58

[9] L.A.Hageman and D.M.Young, *Applied Iterative Methods*, Academic Press, New York, 1981

[10] C.Lanczos, *J.Res.Nat.Bur.Standards*, 49 (1952) 33-53

[11] L.Mansfield, *Comm.Appl.Numer.Methods*, 4 (1988) 151-156

[12] L.Mansfield, *SIAM J.Numer.Anal.*, 27 (1990) 1612-1620

[13] L.Mansfield, *SIAM J.Sci.Statist.Comput.*, 12 (1991) 1314-1323

[14] J.A.Meijerink and H.A.van der Vorst, *Math.Comp.*, 31 (1977) 148-162

[15] S.G.Mulyarchik, S.S.Bielawski, and A.V.Popov, *J.Comput.Phys.*, 110 (1994) 201-211

[16] R.A.Nicolaides, *SIAM J.Numer.Anal.*, 24 (1987) 355-365

[17] J.von Neumann and H.H.Goldstine, *Bull.Am.Math.Soc.*, 53 (1947) 1021-1099

[18] P.Sonneveld, *SIAM J.Sci.Statist.Comput.*, 10 (1989) 36-52

[19] G.W.Stewart, *Numer.Math.*, 21 (1973) 285-297

[20] H.A.van der Vorst, *SIAM J.Sci.Statist.Comput.*, 13 (1992) 631-644

EXTENDED ABSTRACT: PARTIAL ROW PROJECTION METHODS

RANDALL BRAMLEY AND YOUNGHEE LEE
DEPARTMENT OF COMPUTER SCIENCE
INDIANA UNIVERSITY - BLOOMINGTON*

Accelerated row projection (RP) algorithms for solving linear systems $Ax = b$ are a class of iterative methods which in theory converge for any nonsingular matrix. RP methods are by definition ones that require finding the orthogonal projection of vectors onto the null space of block rows of the matrix; see [3, 2, 5] for a general introduction, and a bibliography of the field. The Kaczmarz form, considered here because it has a better spectrum for iterative methods, has an iteration matrix that is the product of such projectors. Because straightforward Kaczmarz method converges slowly for practical problems, typically an outer CG acceleration is applied. Definiteness, symmetry, or localization of the eigenvalues of the coefficient matrix is not required. In spite of this robustness, work has generally been limited to structured systems such as block tridiagonal matrices because unlike many iterative solvers, RP methods cannot be implemented by simply supplying a matrix-vector multiplication routine. Finding the orthogonal projection of vectors onto the null space of block rows of the matrix in practice requires accessing the actual entries in the matrix.

Research on efficient implementation of RP methods has been stalled because of three fundamental problems. The first is computing of the action of the orthogonal projectors on vectors, in particular solving subsystems with coefficient matrix $A_i^T A_i$ where A_i^T is a block row of A. Most practical approachs to date perform a Cholesky factorization $R_i^T R_i$ of $A_i^T A_i$ and then solve two triangular systems when computing the projection $w = P_i d = A_i (R_i^T R_i)^{-1} A_i^T d$. This is practical only if the Cholesky factor is sparse. Although sparsity can be assured by choosing row partitionings that place only a few rows in each block row, the outer CG accelerator requires fewer iterations if the block rows can be made large.

Another difficulty with RP methods is in choosing row partitions. Beyond the need for efficiency in computing the projections, there is no theory to guide the selection of partitionings from the combinatorially large number of possibilities. For centered difference operators on quasi-uniform meshes, [4] gives the tradeoffs in terms of storage and parallelism possible. However, the techniques in that work do not extend to general unstructured meshes for PDE's or other problems.

The third major problem is that even after CG acceleration, RP methods are often too slow. Standard Krylov subspace methods which rely on matrix-vector products with the original matrix A can be made more robust by using an incomplete factorization of A as a preconditioner. Increasing the amount of fill-in (and thus storage required) in the preconditioner often improves the robustness, since in the limiting case the factorization becomes Gaussian elimination. This approach cannot be taken with the RP systems

* WORK SUPPORTED BY NSF GRANTS NSF CDA-9309746, CDA-9303189, AND ASC-9502292

since it is impractical to actually form the RP coefficient matrix. Arioli and Duff [1] have used right preconditioning with a diagonal matrix D. However, this preconditioning is limited to reducing the dependencies between two block rows, and cannot be practically extended to allow incomplete factorizations.

This talk introduces a new "partial" RP algorithm which retains the advantages of RP methods and solves the above three problems. addressed by introducing a new "partial" RP algorithm. When A is partitioned into two block rows, the new system leads to the implicit solution of the auxillary system

$$[A_2^T(I - P_1)A_2]w = b_2 - A_2^T A_1(A_1^T A_1)^{-1}b_1$$

where $P_1 = A_1(A_1^T A_1)^{-1}A_1^T$. A preconditioned CG algorithm is derived, and on each iteration k the approximate solution vector x_k exactly satisfies the first block of equations: $A_i^T x_k = b_i$.

The talk shows how a modification of graph partitioning algorithms allows effective scalable parallelism in all phases of the algorithm, and computation of the required Cholesky factors using $\mathcal{O}(n)$ storage, where A is $n \times n$. Furthermore, standard preconditioning methods such as incomplete factorizations can be used to improve the convergence properties of the partial RP method. Testing results are presented comparing the new partial RP methods with earlier RP methods and Krylov subspace solvers, and showing the effects of different partitionings on the convergence.

REFERENCES

[1] M. Arioli, I. Duff, J. Noailles, and D. Ruiz, *A block projection method for general sparse matrices*, SIAM Journal of Scientific and Statistical Computing, 13 (1992), pp. 47–70.

[2] A. Björck and T. Elfving, *Accelerated projection methods for computing pseudo-inverse solutions of systems of linear equations*, BIT, 19 (1979), pp. 145–163.

[3] R. Bramley and A. Sameh, *Row projection methods for large nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 13 (1992), pp. 168–193.

[4] ——, *Domain decomposition for parallel row projection algorithms*, Applied Numerical Mathematics, 8 (November 1991), pp. 303–315.

[5] C. Kamath and A. Sameh, *A projection method for solving nonsymmetric linear systems on multiprocessors*, Parallel Computing, 9 (1988), pp. 291–312.

GENERALIZED SUBSPACE CORRECTION METHODS

PETTER KOLM† PETER ARBENZ‡ AND WALTER GANDER‡

Extended Abstract
submitted to CMCIM'96 on Iterative Methods, April 9-13, 1996

1. Introduction. A fundamental problem in scientific computing is the solution of large sparse systems of linear equations. Often these systems arise from the discretization of differential equations by finite difference, finite volume or finite element methods. Iterative methods exploiting these sparse structures have proven to be very effective on conventional computers for a wide area of applications. Due to the rapid development and increasing demand for the large computing powers of parallel computers, it has become important to design iterative methods specialized for these new architectures.

We present a class of iterative methods for parallel architectures that evolve from the general framework of subspace correction [7]. The basic ideas go back to the Cimmino [4] and Kaczmarz [6] projection methods, where an iterative process is defined by computing corrections to an approximate solution restricted to subspaces spanned by the rows of the system matrix. These corrections can be performed in succession, as in the Kaczmarz method; or in parallel, as in the Cimmino method. Extensions to these methods to non-overlapping block-rows have been studied, cf. [5, 1, 3, 9, 2], and shown remarkable robustness and potential for parallelism. In this note we focus on extensions to general subspaces, that may overlap, and discuss how the incorporation of weighting schemes can improve convergence significantly.

2. Subspace Correction Methods. For a finite-dimensional vector space $\mathcal{V}$ we consider the linear equation

$$Au = b\,, \quad A \in L(\mathcal{V})\,, \tag{1}$$

with A invertible. Let $(G_i)_{i\in\mathcal{I}}$, $G_i \in L(\mathcal{V})$, $\mathcal{I} \subset \mathbb{N}$, be a finite family of linear maps such that $\mathcal{V} = \bigcup_{i\in\mathcal{I}} \mathcal{R}(G_i)$, where $\mathcal{R}(G_i)$ is the range of G_i. We get by applying these maps to (1)

$$G_iAu = G_ib\,, \quad i \in \mathcal{I}\,, \tag{2}$$

which can be viewed as (1) restricted to the subspace $\mathcal{R}(G_i)$. For an approximate solution u^k of the linear equation (1) we denote by e_i^k the error of the i-th equation of (2). Then with equation (2) we have the *subspace correction equation*

$$G_iAe_i^k = G_ir^k\,,$$

and

$$e_i^k = (G_iA)^\dagger G_ir^k\,,$$

† Computational Mathematics and Mechanics, Royal Institute of Technology, S-100 44 Stockholm, Sweden. email: kolm@nada.kth.se

‡ Institut für Wissenschaftliches Rechnen, Eidgenössische Technische Hochschule, CH-8092 Zürich, Switzerland. email: {arbenz, gander}@inf.ethz.ch

where $r^k := b - Au^k$ and $(G_iA)^\dagger$ is the *Moore-Penrose generalized inverse.* We will consider two natural ways of defining the next iterate u^{k+1}. In the first approach each correction is computed independently and then combined

$$u^{k+1} = u^k + \sum_{i \in \mathcal{I}} E_i e_i^k ,$$

where the weightings E_i are all diagonal and satisfy $\sum_{i \in \mathcal{I}} E_i = I$. Clearly, the j-th component of e_i^k does only have to be computed if the j-th diagonal element of E_i is nonzero. In the second approach, each correction is calculated at a time, thereby using the most recent approximation of the solution, in a Gauss-Seidel type fashion. We remark that the first way explores natural coarse-grained parallelism. In FIG.1 the *generalized parallel* and the *generalized successive subspace correction* methods, *PSC and *SSC, are summarized.

```
/*   *PSC   */
choose (G_i)_{i∈I} and weighting (E_i)_{i∈I}
choose u^0, k = 0
repeat
    for all i do in parallel
        e_i^k = (G_iA)^† G_i r^k
              = (G_iA)^† G_i (b − Au^k)
    end
    u^{k+1} = u^k + Σ_{i∈I} E_i e_i^k
    k = k + 1
until convergence
```

```
/*   *SSC   */
choose (G_i)_{i∈I}
choose u^0, k = 0
repeat
    y^1 = u^k
    for each i do
        e_i = (G_iA)^† G_i (b − Ay^i)
        y^{i+1} = y^i + e_i
    end
    u^{k+1} = y^{m+1}
    k = k + 1
until convergence
```

FIG. 1. *The generalized parallel and the generalized successive subspace correction methods, *PSC and *SSC.*

The *PSC and *SSC algorithms generalize the PSC and SSC algorithms introduced by Xu [9] for symmetric positive definite systems to general linear equations. Here, we also consider an arbitrary family $(G_i)_{i \in \mathcal{I}}$, instead of just orthogonal projections. The weighting incorporated in *PSC has shown to be a useful tool to significantly improve the convergence in applications.

3. Convergence and Consistency. The convergence proofs of the *PSC and *SSC are based upon the following straightforward representation.

PROPOSITION 3.1. **PSC and *SSC are linear stationary methods of first degree, i.e. in the form $u^{k+1} = Qu^k + d$, with $Q_{*PSC} = I - \sum_{i \in \mathcal{I}} E_i (G_iA)^\dagger G_iA$ and $Q_{*SSC} = Q_m \cdots Q_1$ where $Q_i = I - (G_iA)^\dagger G_iA$.*

By using the above representation one can show [7]

THEOREM 3.2. *The *SSC method is convergent and completely consistent, i.e. $\|Q_{*SSC}\| < 1$.*

Showing convergence of the *PSC method needs more care. A proof where the subspaces $\mathcal{R}(G_i)$ are all mutually orthogonal has been established by Elfving [5]. For the general case, incorporating weighting and allowing overlapping subspaces, one can show the following [7].

THEOREM 3.3. *For the consistent weighting $E_i = \frac{1}{m} I$, $i \in \{1, \ldots, m\}$, the *PSC method is convergent and completely consistent.*

Briefly, this is established by constructing an equivalent *SSC algorithm on an extended linear system consisting of the original equation and linear constraints. The result then follows from THEOREM 3.2.

4. Numerical Results. We restrict the experiments to the *PSC method due to its natural parallelism. The implementation is done in C for a 96 node Intel Paragon XP/S5+ at ETH Zürich using message passing primitives from Intel's NX library [8]. As model problem we consider the elliptic partial differential equation

$$-\Delta u + 96\left(x\frac{\partial u}{\partial x} + y\frac{\partial u}{\partial y}\right) = g\,, \quad (x,y) \in (0,1)\times(0,1) \tag{3}$$

with Dirichlet boundary conditions. The differential equation is discretized over a 300×300 grid using centered differences for the first and second order derivatives resulting in a banded linear system. To compare the *PSC implementation with the block-Cimmino method we present computations done where the family $(G_i)_{i\in\mathcal{I}}$ is chosen such that $(G_iA)_{i\in\mathcal{I}}$ constitutes 90 subsequent overlapping block-rows of the banded system matrix. Hereby, the blocks are all equally large and only neighboring blocks are allowed to overlap. The resulting correction equations are solved in minimal norm sense by a sparse LQ decomposition. Four different types of weighting schemes were considered for the correction vectors: in (1) each component is weighted by $\frac{1}{90}$, in (2) overlapping components are halved, in (3) overlapping components are multiplied by weights uniformly distributed between zero and one, and in (4) the components corresponding to half of the overlap in the blocks are set equal to zero making all correction vectors orthogonal.

TABLE 1
*Timings of the LQ decomposition and 1 iteration in seconds for *PSC with different overlap(ov) and weighting scheme 3.*

	Number of Processors				
ov	15	18	30	45	90
0	9.25/0.78	7.64/0.66	4.49/0.43	2.93/0.31	1.40/0.19
2	9.26/0.79	7.66/0.66	4.51/0.43	2.95/0.31	1.39/0.19
4	9.32/0.77	7.70/0.66	4.52/0.43	2.96/0.31	1.40/0.19
10	9.57/0.81	7.96/0.68	4.59/0.45	3.00/0.32	1.40/0.20
20	9.80/0.80	8.12/0.68	4.74/0.44	3.05/0.33	1.43/0.20
40	–	8.45/0.70	4.93/0.45	3.16/0.33	1.44/0.21
80	–	9.22/0.71	5.34/0.47	3.39/0.35	1.49/0.22
120	–	9.89/0.73	5.07/0.49	3.63/0.36	1.55/0.23
160	–	10.58/0.76	6.03/0.51	3.85/0.37	1.59/0.25
200	–	11.35/0.78	6.48/0.53	4.06/0.40	1.63/0.26
280	–	–	6.93/0.56	4.26/0.43	1.73/0.29
360	–	–	7.79/0.59	4.88/0.44	1.84/0.31

TABLE 1 reports the costs for different overlaps, showing that fairly large overlaps are computationally competitive. Computations with so small processor numbers that the whole problem did not fit into main memory have been omitted.

TABLE 2 shows the convergence properties for different weighting and overlap for the model problem. Each test is started with the the same random vector and is stopped if the convergence criterion, $|u_{exact} - u^k|_\infty < 10^{-4}$, is not reached within 5000 iterations. We remark that the *PSC method with and without overlap is superior to the block Cimmino method that does not converge within 5000 iterations. Weighting of type 2–4 improves the convergence significantly.

TABLE 2
*Number of iterations and time in seconds needed for convergence of the *PSC method for different weighting(w) and overlap.*

Overlap				
w	0	40	80	120
none	>5000	–	–	–
1	>5000	>5000	>5000	>5000
2	3913/616.9	4371/712.1	3414/560.6	3135/531.0
3	3913/616.9	4352/708.6	3402/558.6	3053/510.0
4	3913/616.9	4339/706.6	3334/546.9	3251/542.0
Overlap				
w	160	200	280	360
1	>5000	>5000	>5000	>5000
2	3085/521.0	2865/490.8	3232/568.6	3002/542.2
3	3026/511.4	2938/503.1	3145/553.3	2820/509.6
4	2955/498.0	3217/549.1	3211/563.2	2738/490.5

As a stationary method, *PSC in some cases needs too many iterations to achieve convergence. Therefore, it will be important to consider acceleration techniques or to combine the *PSC method with some other iterative process. We believe that these methods are potential preconditioners in parallel environments for standard iterative methods.

REFERENCES

[1] M. ARIOLI, I. DUFF, D. F. RUIZ, AND M. SADKANE, *Techniques for accelerating the block cimmino method*, in Proceedings of the Fifth SIAM Conference on Parallel Processing for Scientific Computing, J. Dongarra, K. Kennedy, P. Messina, D. C. Sorensen, and R. G. Voigt, eds., Philadelphia, PA, 1992, Society for Industrial and Applied Mathematics, pp. 98–104.

[2] M. BENZI, F. SGALLARI, AND G. SPALETTA, *A parallel block projection method of the Cimmino type for finite Markov chains*, in Computations with Markov Chains: Proceedings of the Second International Workshop on the Numerical Solution of Markov Chains, Raleigh, NC, January 16-18, 1995, W. J. Stewart, ed., Kluwer Academic Publishers, 1995, pp. 65–80.

[3] R. BRAMLEY AND A. SAMEH, *Row projection methods for large nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 13 (1992), pp. 168–193.

[4] G. CIMMINO, *Calcolo approssimato per le soluzioni di sistemi di equazioni lineari*, Ric. Sci. Progr. tecn. econom. naz., 9 (1939), pp. 316–333.

[5] T. ELFVING, *Block-iterative methods for consistent and inconsistent linear equations*, Numer. Math., 35 (1980), pp. 1–12.

[6] S. KACZMARZ, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Internat. Acad. Polon. Sci. Lettres, Ser. A, 1937 (1937), pp. 355–357.

[7] P. KOLM, P. ARBENZ, AND W. GANDER, *Generalized subspace correction methods for parallel solution of linear systems*, Tech. Report TRITA-NA-9509, C2M2, Nada, KTH, November 1995. Also published at ETH Zurich, Computer Science Department as Tech. Report No. 241, October 1995. `ftp://ftp.nada.kth.se/Num/publications/trita/trita-na-9509.ps.Z`.

[8] P. PIERCE, *The NX message passing interface*, Parallel Computing, 20 (1994), pp. 463–480.

[9] J. XU, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.

Topic: *Multigrid*	*Session Chair:* *Joel Dendy*	*Room A*
4:45 - 5:15	S. Oliveira	A Multigrid Method for Variational Inequalities
5:15 - 5:45	W. Schmid	A Multigrid Solution Method for Mixed Hybrid Finite Elements
5:45 - 6:15	H.J. Bungartz	A Unidirectional Approach for d-Dimensional Finite Element Methods of Higher Order on Sparse Grids
6:15 - 6:45	M. Brezina	Two-Level Method with Coarse Space Size Independent Convergence

A MULTIGRID METHOD FOR VARIATIONAL INEQUALITIES

S. OLIVEIRA, D.E. STEWART AND W. WU

ABSTRACT. Multigrid methods have been used with great success for solving elliptic partial differential equations. Penalty methods have been successful in solving finite-dimensional quadratic programs. In this paper these two techniques are combined to give a fast method for solving obstacle problems. A nonlinear penalized problem is solved using Newton's method for large values of a penalty parameter. Multigrid methods are used to solve the linear systems in Newton's method. The overall numerical method developed is based on an exterior penalty function, and numerical results showing the peformance of the method have been obtained.

1. INTRODUCTION

Variational inequalities are problems that often arise in connection with free-boundary problems, contact problems and elastic-plastic problems [2]. A well-known example of such a problem is the so-called "obstacle problem" [2, pp. 104ff]. The idea is to model an elastic sheet which is suspended over some terrain. The sheet is assumed to be supported at its edges, but it might also be supported by the ground, though the region the sheet is in contact with the ground is unknown. Where the sheet is suspended, the usual equations apply, though over the region of contact, a frictionless force is exerted on the sheet by the ground to keep it from sinking into the ground. A linearized version of this can be represented by the equations below, where $u(x,y)$ is the downward vertical displacement of the sheet at co-ordinates (x,y).

$$\begin{aligned} -\Delta u &= c(x,y) \qquad \text{where } u(x,y) > b(x,y) \\ -\Delta u &\geq c(x,y) \qquad \text{where } u(x,y) = b(x,y) \end{aligned} \tag{1}$$

where $b(x,y)$ is the downwards displacement of the ground and $c(x,y)$ is the downward force applied per unit area of the sheet. Typically this force would be gravity. Such problems can be considered to be infinite-dimensional versions of quadratic programs as they can be reformulated as minimization problems in Hilbert spaces. For example, the above obstacle problem can be reformulated as

$$\min_{u \geq b} \int_\Omega \left[\frac{1}{2} |\nabla u|^2 - c(x)u \right] dV. \tag{2}$$

Numerically such methods can be discretized and solved as large quadratic programs. However, in spite of the recent advances in such methods, the guaranteed convergence rate of such methods are highly dependent on the dimension of the discretized problem. Furthermore, the rates of convergence go to zero as the

dimension is increased. Here an alternative approach is to apply multigrid to a grid-independent penalty formulation by developing algorithms in the appropriate Hilbert space. The convergence rates of these methods will approach that of the infinite-dimensional algorithm as the grid is refined and the approximations are made more accurate [1]. To achieve full mesh-independence, the multigrid method must be modified to properly deal with large penalty parameters. The ideal method is known to be monotone for appropriate penalty functions [4] so that the successive Newton iterates $u^{(k+1)}(x)$ are monotone in k for every x. The corresponding discrete property holds provided a lumped mass scheme is used for the penalty terms, and the multigrid method is monotone when applied to M-matrices.

The methods described here are believed to be competitive with the best published algorithms which are essentially active set methods (see Kornhuber [3]).

2. Theory and numerical properties

The basic problem to be solved here is

$$\min_{u \geq b} J(u) = \int_\Omega \left[\frac{1}{2} |\nabla u|^2 - c(x) u \right] dV \tag{3}$$

over $u \in H_0^1(\Omega)$. We assume that b is Lipschitz continuous and $b \leq 0$ on $\partial\Omega$, and that c belongs to $L^2(\Omega)$. The region Ω is assumed to be an open subset of $\mathbb{R}^n$ with a Lipschitz boundary, and that $n \leq 4$. This minimization problem is equivalent to the variational inequality

$$\int_\Omega [\nabla u \cdot \nabla(v-u) - c \cdot (v-u)]\, dV \geq 0, \qquad \text{for all } v \in H_0^1(\Omega) \text{ where } v \geq b, \text{ and } u \geq b. \tag{4}$$

This in turn is equivalent to the pointwise inequalities in (1).

A solution exists for (3) since the functional J is a proper convex and norm-continuous function and is coercive:

$$J(u) \geq \|u\|_{H_0^1}^2 - \|c\|_{H^{-1}} \|u\|_{H_0^1}.$$

Note that here we use the norm

$$\|u\|_{H_0^1} = \sqrt{\int_\Omega |\nabla u|^2 \, dV}.$$

Thus J is weakly lower semi-continuous (LSC), and by weak compactness of bounded sets in H_0^1 and coercivity, minima exist. Strict convexity ensures uniqueness and continuous dependence on the data of the problem.

Since (3) is a difficult problem to solve directly, we consider a penalty functional

$$J_\beta(u) = \int_\Omega \left[\frac{1}{2} |\nabla u|^2 - c \cdot u + \beta \Phi(x,u) \right] dV. \tag{5}$$

Here we use the *exterior penalty* function $\Phi(x,u) = (u - b(x))_+^2$, and to obtain the solution of the original variational inequality we need to find the limit of minimizers

u_β of J_β as $\beta \to \infty$. The exact form of Φ is not important, but the following properties are:

$$\text{there is an } h \in L^1(\Omega) \text{ where } \Phi(\cdot, u) \geq h \text{ for all feasible } u.$$

$$\frac{\partial \Phi}{\partial u}, \frac{\partial^2 \Phi}{\partial u^2} \geq 0. \tag{6}$$

$$\frac{\partial^2 \Phi}{\partial u^2} \text{ is non-decreasing in } u.$$

The necessary conditions for a minimum of J_β are that

$$-\Delta u_\beta - c + \beta \phi(x, u_\beta) = 0 \tag{7}$$

in Ω and $u_\beta = 0$ on $\partial\Omega$, where $\phi(x,u) = (\partial\Phi/\partial u)(x,u)$. This is a nonlinear equation for u_β. This is solved by means of Newton's method. At each stage of Newton's method, with $u^{(k)}$ being the approximation at the k iteration, the correction $w^{(k)} = u^{(k+1)} - u^{(k)}$ satisfies the equation

$$-\Delta(u^{(k)} + w^{(k)}) - c + \beta\phi(x, u^{(k)}) + \beta \frac{\partial \phi}{\partial u}(x, u^{(k)}) w^{(k)} = 0 \tag{8}$$

in Ω and $w^{(k)} = 0$ on $\partial\Omega$.

The Newton equations (8) can be rewritten as

$$-\Delta w^{(k)} + \beta \frac{\partial \phi}{\partial u}(x, u^{(k)})\, w^{(k)} = f^{(k)} \tag{9}$$

where $f_k = -[-\Delta u^{(k)} - c + \beta\phi(x, u^{(k)})]$. Note that $u^{(k)} \in H_0^1$, so $\Delta u^{(k)} \in H^{-1}$; $u^{(k)}$ may still be unbounded (for $n \geq 2$), so $\phi(\cdot, u^{(k)})$ and $(\partial\phi/\partial u)(\cdot, u^{(k)})$ may also be unbounded. This can cause problems with the existence theory for these PDE's. However, solutions can still be shown to exist by considering the following strictly convex, coercive quadratic functional

$$K_{\beta,u,f}(w) = \int_\Omega \left[\frac{1}{2}|\nabla w|^2 + \frac{1}{2}\beta \frac{\partial \phi}{\partial u}(x,u)\, w^2 - f \cdot w \right] dV. \tag{10}$$

Note that $w^{(k)}$ solves (9) if and only if it minimizes $K_{\beta,u^{(k)},f^{(k)}}$. By the same arguments that are used for the existence of u_β, and the solution of the variational inequality, a unique solution $w = w^{(k)}$ exists which solves (9).

Newton's method typically exhibits only local convergence. Here, however, it is possible to show that the convergence is monotone provided that $-\Delta u^{(0)} - c + \beta\phi(x, u^{(0)}) \geq 0$ everywhere in Ω; then, $u^{(0)} \geq u^{(1)} \geq u^{(2)} \geq \cdots \geq u_\beta$. This is described in some detail in the functional analytic context in [4]. This is essentially due to the fact that $(-\Delta)^{-1}$ is a monotone operator for Dirichlet boundary conditions. Obtaining a starting solution $u^{(0)}$ where $-\Delta u^{(0)} - c + \beta\phi(x, u^{(0)}) \geq 0$ is not difficult; one way to do this is to solve $-\Delta u^{(0)} - c = 0$. Then as $\phi(x,u) \geq 0$ for any value of u, the desired inequality holds for this choice of $u^{(0)}$.

In Stewart and Wright [4] quadratic convergence for this monotone Newton method is shown provided $\partial^2\phi/\partial u^2$ is uniformly bounded. However, for $\Phi(x,u) = (b(x) - u)_+^2$, there is no second derivative with respect to u where $u = b(x)$.

3. Numerical solution

For the test problems, the standard five point difference stencil is used in two dimensions. Note that this is equivalent to the piecewise linear finite element on a regular triangular mesh based on a regular square grid.

The choice of discretization can be important in order to preserve the monotonicity properties of the Newton iteration, which require the property that the discrete operator $-\Delta_h$ is an M-matrix. If, for example, higher order finite elements are used, this property is lost, and the discrete Newton method cannot expected to be monotone. At least for piecewise linear finite element methods, $-\Delta_h$ is an M-matrix.

To solve the penalized problem for the variational inequality, a Newton–multigrid method was used. This involves setting up the linearized equations for the discrete system, and applying a multigrid method to the linearized system. The algorithm used is a modification of the FMV multigrid method. The modification uses Gauss–Seidel for the relaxation as the original method does, but with the diagonal penalty terms added. On coarse grids, the problem is to properly define the diagonal penalty terms. For a given coarse grid node, the diagonal penalty term is defined in terms of the value of u at the corresponding fine grid node. For the penalty function $\Phi(x,u) = (b(x)-u)_+^2$, the diagonal penalty term is 2 if $u \le b(x)$, but zero otherwise. This may be inappropriate where the geometry of the contact region is complex, as a node of the coarse grid may be close to a contact region, but because it is not itself in the contact region, the penalty term is zero. Nevertheless, this method is often able to solve the penalized problem for large values of β.

4. Numerical results

The first test problem was the problem of a semi-sphere as an obstacle, which was chosen because it was possible to obtain an exact solution. The actual obstacle had the form

$$b(x,y) = \begin{cases} \sqrt{1-x^2-y^2} & x^2+y^2 \le 1 \\ -\infty & \text{otherwise} \end{cases}.$$

The obstacle problem was solved with $c = 0$ on $\Omega = (-2,+2) \times (-2,+2)$ with Dirichlet boundary conditions consistent with the exact solution

The most important issues for this method is the dependence of the number of iterations needed on the grid spacing and the dependence on the penalty parameter β. Ideally, the number of iterations would be bounded, or only slowly growing, as the grid spacing becomes small and the penalty parameter becomes large.

Table 1 shows the number of inner iterations (calls to the multigrid routine) and the number of outer iterations (the number of Newton steps) for different grid sizes for the semi-sphere obstacle. This was solved on the square $\Omega = (-2,+2)\times(-2,+2)$ with inhomogeneous boundary conditions consistent with the exact solution

$$u(x,y) = \begin{cases} \sqrt{1-r^2} & r \le r^* \\ -(r^*)^2 \ln(r/R)/\sqrt{1-(r^*)^2} & r \ge r^* \end{cases}$$

where $r = \sqrt{x^2+y^2}$, $R = 2$, and r^* satisfies

$$(r^*)^2 \left[1 - \ln(r^*/R)\right] = 1.$$

β	grid size	# outer loops	# inner loops
10^3	32×32	5	10
10^3	64×64	9	16
10^3	128×128	12	20

TABLE 1. Dependence on grid size (semi-sphere obstacle)

β	grid size	# outer loops	# inner loops
10^2	64×64	5	8
10^3	64×64	9	16
10^4	64×64	8	17
10^5	64×64	8	18

TABLE 2. Dependence on penalty parameter β (semi-sphere obstacle)

With $R = 2$, this gives $r^* = 0.6979651482\ldots$. A plot of the numerically computed solution is shown in Figure 1. The convergence history for the algorithm applied to this problem with various grid spacings is shown in Figure 2. A plot of the errors in the computed solution for the 32×32 grid with $\beta = 10^3$ is shown in Figure 3. As can be seen in this figure, the solution is very accurate in the region of contact, but is least accurate just outside. This is probably due to the jump discontinuity in the contact forces, which is difficult to resolve accurately without adaptive methods.

The penalty parameter β was set to 10^3. The stopping tolerance for the inner loop was 10^{-2}, and the stopping tolerance for the outer loop was 10^{-3}.

The convergence history for the algorithm applied to this problem with various penalty parameters is shown in Figure 4.

Table 2 shows the number of inner and outer loops for different values of the penalty parameter β for the semi-sphere obstacle using the same stopping tolerances as before, and a 64×64 grid.

REFERENCES

1. E.L. Allgower, K. Böhmer, F.A. Potra, and W.C. Rhenboldt. A mesh-independent principle for operator equations. *SIAM J. Numer. Anal.*, 23:160–169, 1986.
2. C. Baiocchi and A. Capelo. *Variational and Quasivariational Inequalities: Applications to Free Boundary Problems.* Wiley, Chichester, New York, 1984.
3. R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities. I. *Numer. Math.*, 69(2):167–184, 1994.
4. D.E. Stewart and S.J. Wright. Monotone convergent methods for a variational inequality. In *Proceedings Miniconference on Nonlinear Analysis*, volume 33, pages 195–208. CMA, School of Mathematical Sciences, Australian National University, 1994. Miniconference held Brisbane, Australia, 1993.

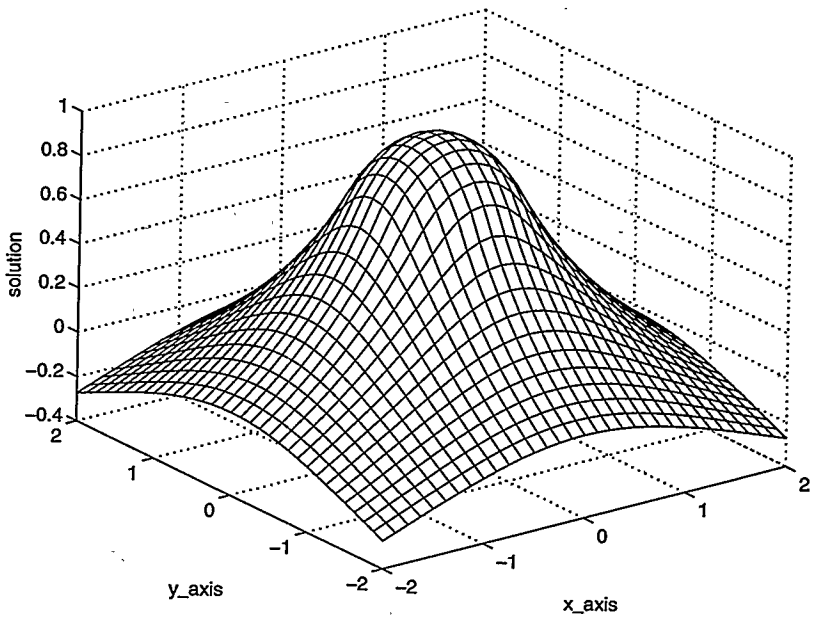


FIGURE 1. Numerical solution for semi-sphere obstacle

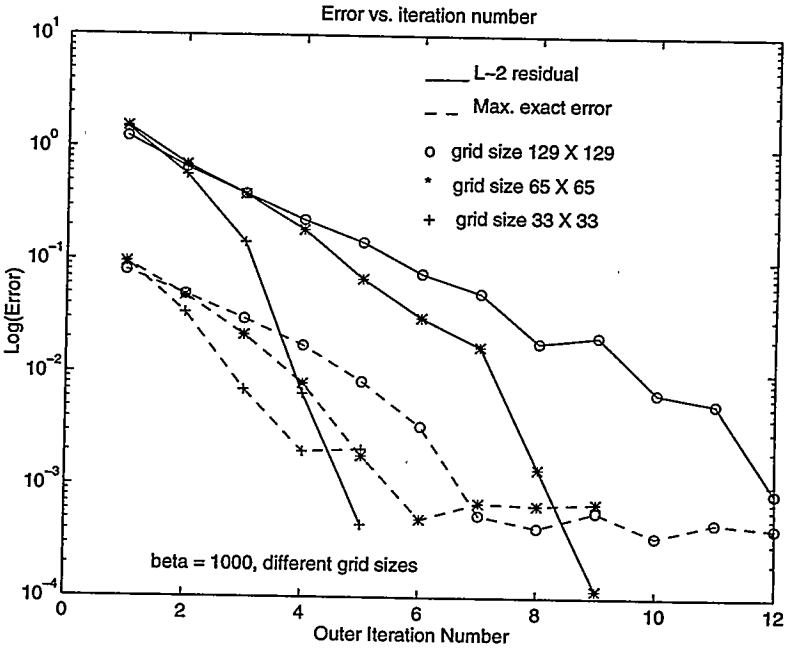


FIGURE 2. Convergence history for different grid sizes (semi-sphere obstacle)

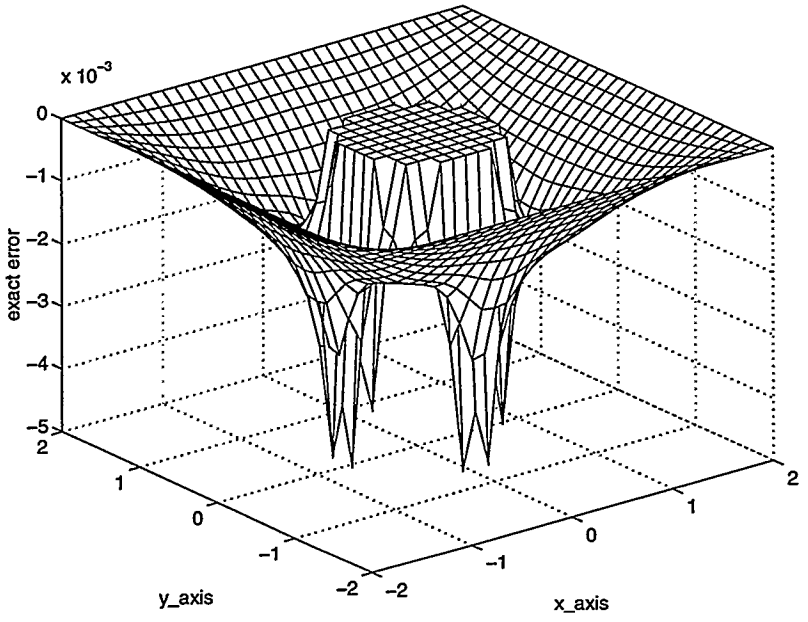


FIGURE 3. Errors in computed solution: semi-sphere obstacle on 32×32 grid and $\beta = 10^3$

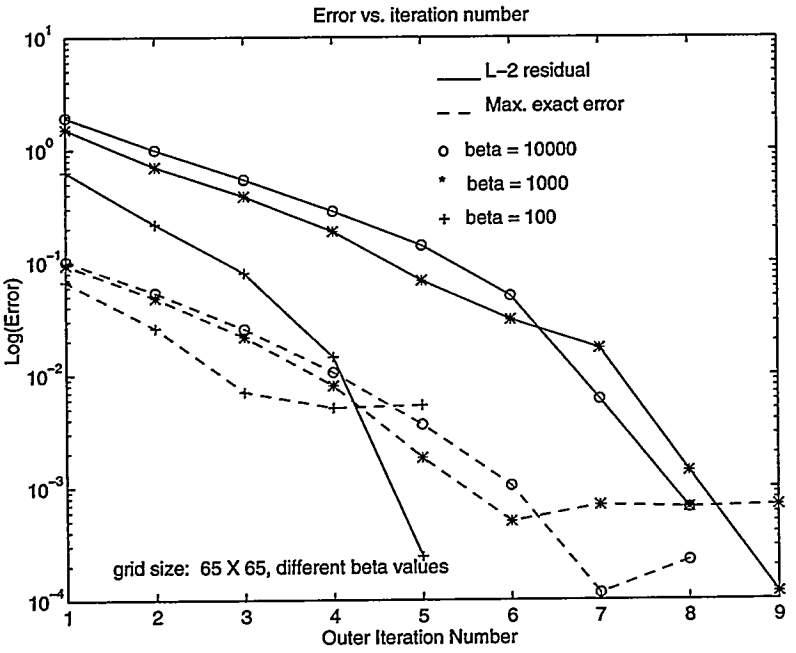


FIGURE 4. Convergence history for different β (semi-sphere obstacle)

A MULTIGRID SOLUTION METHOD FOR MIXED HYBRID FINITE ELEMENTS

WERNER SCHMID*

Summary. We consider the multigrid solution of linear equations arising within the discretization of elliptic second order boundary value problems of the form

$$L(u) := -\nabla(A\nabla u) + bu = f \text{ in } \Omega \in \mathbb{R}^3, \quad u = 0 \text{ on } \Gamma,$$

by mixed hybrid finite elements. Using the equivalence of mixed hybrid finite elements and non-conforming nodal finite elements (cf. Arbogast/Chen [1]), we construct a multigrid scheme for the corresponding non-conforming finite elements, and, by this equivalence, for the mixed hybrid finite elements, following guidelines from Arbogast/Chen [1].
For a rectangular triangulation of the computational domain, this non-conforming schemes are the so-called nodal finite elements (cf. Hennart/Del Valle [6]). We explicitly construct prolongation and restriction operators for this type of non-conforming finite elements. We discuss the use of plain multigrid and the multilevel-preconditioned cg-method and compare their efficiency in numerical tests.

Key words. Mixed Finite Elements, Hybridization, Iterative Methods, Multigrid.

AMS(MOS) subject classifications. 65N55, 65N30, 82D75

1. Discretization. We consider in the sequel the typical second order elliptic boundary value problem

$$\begin{aligned} L(u) := -\mathrm{div}(A\ \mathbf{grad} u) + bu = f \text{ in } \Omega \in \mathbb{R}^3, \\ u = 0 \text{ on } \Gamma := \partial\Omega, \end{aligned} \tag{1}$$

where A is a matrix-valued, positive definite function on Ω, i.e., there is a constant $\alpha > 0$ with $\sum_{i,j=1,2} a_{i,j}(x)\xi_i\xi_j \geq \alpha||\xi||^2$ for all $x \in \Omega$, $\xi \in \mathbb{R}^d$. We assume f and $b \in L^2(\Omega)$ and $b \geq 0$. We assume further that the coefficients are smooth enough to ensure the required regularity of u, whenever necessary.

For applications where both u and the (possibly weighted) gradient $A\ \mathbf{grad} u$ are of interest, the use of mixed finite elements is a reasonable choice. Then equation (1) is formally written as first order system

$$\begin{aligned} A^{-1}\mathbf{j} + \mathbf{grad} u = 0 \\ \mathrm{div}\mathbf{j} + bu = f \end{aligned} \tag{2}$$

with the corresponding boundary conditions. For simplicity we assume homogeneous Dirichlet boundary conditions throughout.

* Mathemat.-Naturwiss. Fakultät, Universität Augsburg, D-86159 Augsburg, Germany, (schmid@math.uni-augsburg.de). Partially supported by a Ph.D. grant of SIEMENS AG and grant 03-H07TUM of the German Federal Department of Research and Technology (BMFT)

Let us define $V = H(\mathrm{div};\Omega) := \{\mathbf{q} \in (L_2(\Omega))^d; \mathrm{div}\mathbf{q} \in L^2(\Omega)\}$.
The variational formulation of (2) is then given by

$$\begin{aligned} \overbrace{(A^{-1}\mathbf{j},\mathbf{q})}^{a(\mathbf{j},\mathbf{q})} - \overbrace{(u,\mathrm{div}\mathbf{q})}^{-c(u,\mathbf{q})} &= 0, \quad \mathbf{q} \in V \\ \underbrace{(w,\mathrm{div}\mathbf{j})}_{-c(w,\mathbf{j})} + \underbrace{(bu,w)}_{b(u,w)} &= \underbrace{(f,w)}_{f(w)}, \quad w \in W \end{aligned} \tag{3}$$

For numerical computations we have to use adequate finite dimensional function spaces. In the case of the saddle-point problem above, these function spaces have to be chosen carefully in order to guarantee existence and uniqueness of the solution in the discrete case. This can, e.g., be accomplished by fulfilling the discrete Babuška-Brezzi-condition, cmp. Brezzi / Fortin [4, Kap. II.2].

One possible choice in 3D (for hexaedral grid), the one we use in the sequel, is the use of the so-called Raviart-Thomas-Nédélec spaces of lowest order as discrete substitute of $H(\mathrm{div};\Omega)$ (cmp. Nédélec [7, 8]) and the corresponding discrete space for the primal variable u: Starting from a domain $\Omega \in \mathbb{R}^3$ and a coarse grid $\mathcal{T}_0$ consisting of cubes K with radius h, we have the following conforming discretization

$$\begin{aligned} V_h = RT_{[0]}(\Omega;\mathcal{T}_h) &:= \{\mathbf{q} \in H(\mathrm{div};\Omega); \mathbf{q}|_K \in RT_{[0]}(K)\} \\ W_h = W_{[0]}(\Omega;\mathcal{T}_h) &:= \{w \in W(\equiv L^2(\Omega)); w|_K \in Q_0(K)\} \end{aligned}$$

with $RT_{[0]}(K) := Q_{1,0,0} \times Q_{0,1,0} \times Q_{0,0,1}$.
Here, we use $Q_{k,l,m} := \{c_{\alpha,\beta,\gamma} x^\alpha y^\beta z^\gamma;\ 0 \le \alpha \le k;\ 0 \le \beta \le l;\ 0 \le \gamma \le m;\}$ and $Q_k = Q_{k,k,k}$.

Now we can switch from (3) to the finite dimensional problem

$$\begin{aligned} (A^{-1}\mathbf{j_h},\mathbf{q_h}) - (u_h,\mathrm{div}\mathbf{q_h}) &= 0, \quad \mathbf{q_h} \in RT_{[0]}(\Omega;\mathcal{T}_h) \\ (w_h,\mathrm{div}\mathbf{j_h}) + (bu_h,w_h) &= (f,w_h), \quad w_h \in W_{[0]}(\Omega;\mathcal{T}_h) \end{aligned} \tag{4}$$

$V_h \times W_h$ fulfills the inf-sup-condition. Therefore, existence and uniqueness of the discrete solution of (4) is guaranteed. Furthermore, this solution converges to the solution of (3) with $h \to 0$.

Every vector-field $v_h \in V_h$ must satisfy (like every $v \in V \equiv H(\mathrm{div};\Omega)$) the condition that its normal component is continuous. This leads to a coupling between the elements $K \in T_0$ via

$$\int_{\Gamma_K} \mathbf{n} \cdot \mathbf{q}_h d\sigma = -\int_{\Gamma_{K'}} \mathbf{n} \cdot \mathbf{q}_h d\sigma \tag{5}$$

for the common faces of two neighbour elements K, K'. In operator (or matrix) notation with
$A : RT_{[0]}(\Omega;\mathcal{T}_h) \to RT_{[0]}(\Omega;\mathcal{T}_h)^*$, $C : RT_{[0]}(\Omega;\mathcal{T}_h) \to W_{[0]}(\Omega;\mathcal{T}_h)^*$ and

$B : W_{[0]}(\Omega;\mathcal{T}_h) \to W_{[0]}(\Omega;\mathcal{T}_h)^*$, we arrive at the following, uniquely solvable, but indefinite linear system of equations with global coupling

$$\begin{pmatrix} A & C^T \\ C & -B \end{pmatrix} \begin{pmatrix} \mathbf{j}_h \\ u_h \end{pmatrix} = \begin{pmatrix} 0 \\ -f \end{pmatrix}.$$

Iterative solution of those indefinite problems is usually harder to do than for positive definite systems. Therefore, we transform (4) into a symmetric positive definite (s.p.d.) system by the technique of *hybridization*. Then a subsequent post-processing gives the solution of the original problem.

The main idea of hybridization is to eliminate continuity of the normal components (5) from the function space, i.e.,

$$RT_{[0]}(\Omega;\mathcal{T}_h) := \{q \in H(\mathrm{div};\Omega); \mathbf{q}|_K \in RT_{[0]}(K); K \in \mathcal{T}_h\}$$

$$\Downarrow$$

$$RT^{-1}_{[0]}(\Omega) := \{\mathbf{q}|_K \in RT_{[0]}(K); K \in \mathcal{T}_h\}$$

Now we can use function spaces for the vector-field that are locally defined, i.e., there will be no inter-element coupling for the vector valued variable $\mathbf{j_h}$.
Continuity of the normal components is assured by introducing a further equation:

$$\begin{aligned} \hat{a}(\mathbf{j}_h, \mathbf{q}_h) + \hat{c}(\mathbf{q}_h, u_h) &= -\hat{d}(\mu_h, \mathbf{q}_h) \quad , \mathbf{q}_h \in RT^{-1}_{[0]}(\Omega;\mathcal{T}_h), \\ \hat{c}(\mathbf{j}_h, v_h) - \hat{b}(u_h, v_h) &= -f(v_h) \quad , v_h \in W_{[0]}(\Omega;\mathcal{T}_h) \\ \hat{d}(\rho_h, \mathbf{j}_h) &= 0 \quad , \rho_h \in M_{[0]}(\Omega;\mathcal{F}_h), \end{aligned} \tag{6}$$

The bilinear forms are locally defined on single elements wherever necessary:

$$\hat{a}(\mathbf{p}_h, \mathbf{q}_h) = a(\mathbf{p}_h, \mathbf{q}_h),\ \hat{c}(\mathbf{p}_h, v_h) = \sum_{K\in\mathcal{T}_h} (v_h, \mathrm{div}\mathbf{q}_h)|_K,$$

$$\hat{b}(v_h, w_h) = (bv_h, w_h) \text{ (in } \Omega), \text{ and } \hat{d}(\rho_h, \mathbf{q}_h) := \sum_{K\in\mathcal{T}_h} \int_K \rho_h \mathbf{n}\cdot\mathbf{q_h} d\sigma \text{ with}$$

$$M_{[0]}(\Omega;\mathcal{F}_h) := \{\rho_h \in (L^2(\mathcal{E}_h))^3; \rho_h|_e \in P_0(e); e \in \mathcal{E}_h \cap \Omega; \rho_h|_e = 0, e \in \mathcal{E}_h \cap \Gamma_D\}.$$

It seems that we have inserted further unknowns requiring more work than before. But since $\mathbf{j}_h$ is defined element by element it can a priori be eliminated leaving us with a linear system of equations in u_h and λ_h only that even is s.p.d. Using matrix notation and starting with

$$\begin{pmatrix} \hat{A} & \hat{C}^T & \hat{D} \\ \hat{C} & -\hat{B} & 0 \\ \hat{D} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{j}_h \\ u_h \\ \lambda_h \end{pmatrix} = \begin{pmatrix} 0 \\ -f \\ 0 \end{pmatrix} \tag{7}$$

we have $\mathbf{j}_h = -\hat{A}^{-1}(\hat{C}u_h + \hat{D}\lambda_h)$. Using this relation, we get

$$\begin{pmatrix} \hat{C}\hat{A}^{-1}\hat{C}^T + \hat{B} & \hat{C}\hat{A}^{-1}\hat{D}^T \\ \hat{D}\hat{A}^{-1}\hat{C}^T & \hat{D}\hat{A}^{-1}\hat{D}^T \end{pmatrix} \begin{pmatrix} u_h \\ \lambda_h \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{8}$$

A local a priori elimination of u_h leads to

$$D\hat{A}^{-1}\hat{C}^T[\hat{C}\hat{A}^{-1}\hat{C}^T+\hat{B}]^{-1}\hat{C}\hat{A}_{-1}\hat{D}^T-\hat{D}\hat{A}^{-1}\hat{D}^T=D\hat{A}^{-1}\hat{C}^T[\hat{C}\hat{A}^{-1}\hat{C}^T+\hat{B}]^{-1}f. \quad (9)$$

REMARK 1.1. *We will not perform this second elimination and refer to Arbogast/Chen [1] and Arnold/Brezzi [2] for this linear system.*
In the application we have in mind (i.e., neutron diffusion), u_h cannot be eliminated because it explicitly is used in the right hand side. Therefore, we further deal with the system of equations in u_h and λ_h.
A system in λ_h only would probably have better numerical properties since the use degrees of freedom defined on faces and in the interior of an element usually leads to a worse condition number than just the use of degrees of freedom defined on faces.

REMARK 1.2. *The solution $(\mathbf{j}_h, u_h, \lambda_h)$ of the mixed hybrid finite element ansatz always exists and is unique. If $(\mathbf{j}_h^*, u_h^*)$ is the solution of the mixed finite element ansatz (4), then we have $\mathbf{j}_h \equiv \mathbf{j}_h^*$ and $u_h \equiv u_h^*$. For a proof see e.g. Arnold/Brezzi [2]. A general analysis can be found in Brezzi/Fortin[4, Chapter V.1.2].*

The relation between mixed hybrid finite elements and a corresponding non-conforming primal approach is given by

THEOREM 1.1. **(Equivalence of mixed hybrid and non-conforming FE)** *We assume piecewise constant coefficients and a diagonal A in (1). Further let $\tilde{u}_h \in N(\Omega;\mathcal{T}_h)$ be a non-conforming approximation of the solution u of (1) defined by the mixed hybrid solution $(\mathbf{j}_h, u_h, \lambda_h)$ for (1) via $\Pi_h^l(u-\tilde{u}_h^*)=0$ and $P_h^l(u-\tilde{u}_h^*)=0$. Then $\tilde{u}_h$ is equivalent to a non-conforming primal approximation $\hat{u}_h \in N(\Omega;\mathcal{T}_h)$ satisfying $\hat{a}_h(\hat{u}_h, v_h)=\hat{f}(v_h)\ \forall v_h \in N(\Omega;\mathcal{T}_h)$ where $\hat{f}(v_h)=\sum_{K\in\mathcal{T}_h}\int_K P_h^l f v_h dx\ \forall v_h \in N(\Omega;\mathcal{T}_h)$ and $\hat{a}_h(.,.)$ is defined by*

$$\hat{a}_h(w_h, v_h) := \sum_{K\in\mathcal{T}_h}\int_K A\ \mathbf{grad}v_h \cdot Q_0^h \mathbf{grad}w_h dx + \int_K v_h P_h^l b w_h dx. \quad (10)$$

For a proof and the explicit definition of the non-conforming ansatz space see Hennart /Del Valle [6, Chapter V].

2. Multilevel methods. Starting point of our multilevel method is the discretization of (1) using the mixed hybrid ansatz (or of the corresponding non-conforming finite element ansatz, resp.). We assume a triangulation $\mathcal{T}_0$ for $\Omega \subset \mathbb{R}^3$ with radius h_0 to be given. For $k \geq 1$ the triangulation $\mathcal{T}_k$ is then given by uniform refinement of $\mathcal{T}_{k-1}$ which results in $h_k := 2^{-k}\cdot h_0$. On the whole, we have a family $(\mathcal{T}_k)_{k\geq 0}$ of triangulations with a corresponding family $(N(\mathcal{T}_k;\Omega))_{k\geq 0}$ of non-conforming ansatz spaces and a family of non-conforming bilinear forms $(a_k(.,.))_{k\geq 0}$ according to the non-conforming ansatz equivalent to the mixed hybrid finite elements. We have $N(\Omega;\mathcal{T}_{l-1}) \not\subset N(\Omega;\mathcal{T}_l)$, i.e., there is *no* natural injection operator from $N(\Omega;\mathcal{T}_{l-1})$ into $N(\Omega;\mathcal{T}_l)$. Therefore, in analogy to Arbogast/Chen [1, Kapitel 8], we introduce an interpolation operator

$I_{k-1}^k : N(\Omega; \mathcal{T}_{l-1}) \longrightarrow N(\Omega; \mathcal{T}_l)$. In our case a further condition dealing with the inner degrees of freedom u_h is necessary. I_{k-1}^k is defined for all $\xi \in N(\Omega; \mathcal{T}_{l-1})$ by

$$\frac{1}{|K|}\int_K I_{k-1}^k \xi \, dx := \frac{1}{|K|}\int_K \xi \, dx \; \forall K \in \mathcal{T}_k \qquad \text{and} \tag{11}$$

$$\frac{1}{|f|}\int_f I_{k-1}^k \xi \, ds := \begin{cases} \frac{1}{2|f|}\int_f (\xi|_{K_1} + \xi|_{K_2}) \, ds, & f \subset K_1 \cap K_2;\ K_1, K_2 \in \mathcal{T}_{k-1} \\ \frac{1}{|f|}\int_f \xi \, ds, & f \not\subset \partial K', K' \in \mathcal{T}_{k-1} \\ 0, & f \subset \Gamma_D \end{cases} \tag{12}$$

The operator I_{k-1}^k can be calculated easily using the local basis functions of the nonconforming ansatz (cf. Hennart [5, Appendix C]) .
The multigrid algorithm is then given by (Correction-Scheme):

Level $k = 0$ $u_0^1 \approx A_0^{-1} f_0$ calculated using another method,
e.g., a direct method as LU-decomposition.
Level $k \geq 1$ Pre-smoothing: $u_k^0 \leftarrow S^{\nu_1}(k, u_k^0, f_k)$
Calculation of the residual: $r_k = f_k - A_k \cdot u_k^0$
Restriction of the residual: $r_{k-1} = I_k^{k-1} r_k$
Calculation of a correction on level k-1:
$u_{k-1}^0 = 0$
for $j = 1, \ldots, p : u_{k-1}^0 \leftarrow MG(k-1, u_{k-1}^0, r_{k-1})$
Prolongation of the correction: $v_k = I_{k-1}^k u_{k-1}^0$
Coarse grid correction: $u_k^0 = u_k^0 - v_k$

The restriction operator I_k^{k-1} is given as the transposed of the prolongation operator: $I_k^{k-1} := (I_{k-1}^k)^T$. For this scheme the following theorem can be proved:

THEOREM 2.1. **(Multigrid convergence of W-cycle)**
For the W-cycle, a constant $0 \leq \gamma < 1$ and a sufficient number $m \geq 1$ of smoothing steps exists, both independent of the level k, such that we have

$$||u_k - MG(k, u_k^0, f_k)||_k \leq \gamma ||u_k - u_k^0||_k \tag{13}$$

for the vector u_k^0, where u_k is the solution of (8) (or of the equivalent nonconforming system, resp.).

Proof: A proof for a system of equations in λ_h only can be found in Arbogast/Chen [1, Appendix]. The proof in our case is a simple variation where only the extension of the prolongation operator for the unknown λ_h to the case of unknowns (u_h, λ_h) must be taken into account. Details can be found in Schmid [9, Chapter 4.3]. □

3. Numerical Results. As test problems, we use Helmholtz equations of the form

$$-\text{div}(D \; \textbf{grad} u(x,y,z)) + c \cdot u(x,y,z) = f(x,y,z) \text{ in } \Omega$$

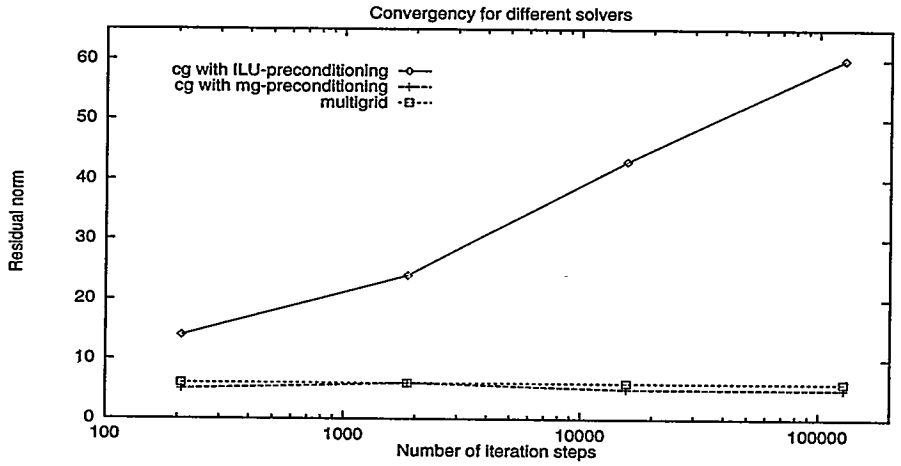


FIG. 1. *Number of iteration steps, different levels,* $u_1(x, y, z)$

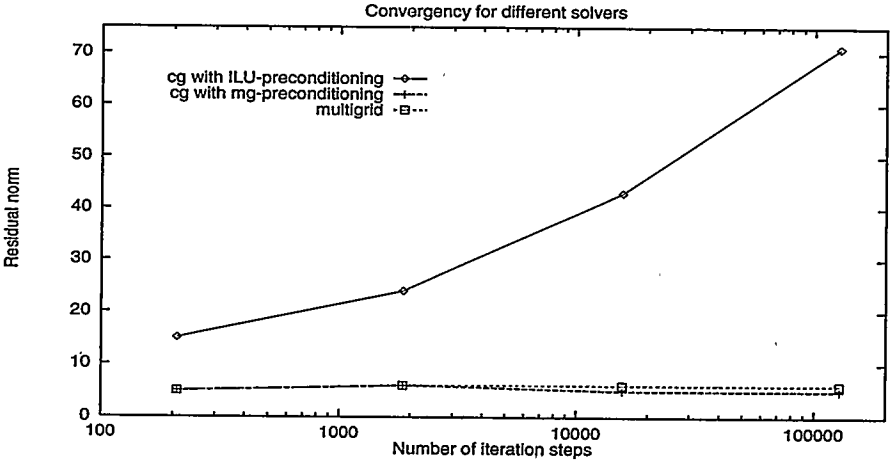


FIG. 2. *Number of iteration steps, different levels,*$u_2(x, y, z)$

with $\Omega := [0,1]^3$, homogeneous Dirichlet boundary conditions and different right hand sides and coefficients. The analytical solutions were given by

$$
\begin{aligned}
u_1(x,y,z) := \;& 1000 \cdot exp(-100 \cdot (x-0.4)^2 - (y-0.2)^2 - (z-0.3)^2) \\
& \cdot\{x\ (x-1.0)\ y\ (y-1.0)\ z\ (z-1.0)\} \\
u_2(x,y,z) := \;& 1000 \exp(-100 \cdot [(x-0.4)^2 - (y-0.2)^2]) \\
& \cdot\{x(x-1.0)y(y-1.0)z(z-1.0)\} \\
u_3(x,y,z) := \;& -\tfrac{1}{D} \cdot 1000x(x-1.0)(x-0.5)y(y-1.0)(y-0.5)z(z-1.0)(z-0.5) \\
& \text{with } D = 1/50 \text{ in } [0.5,1.0]^3,\ D = 1.0 \ \text{ elsewhere.}
\end{aligned}
$$

In our numerical results, we compare three different iterative solvers: cg-iteration with ILU-preconditioning, cg-iteration with one multigrid step as preconditioner and plain multigrid as described above.
We start with a coarse grid of 20 unknowns and refined uniformly resulting in a grid with 128000 unknowns on level 4. Calculations for Fig. 1-3 have been performed using 2 pre- and 2 postsmoothing steps with ILU-smoothing.

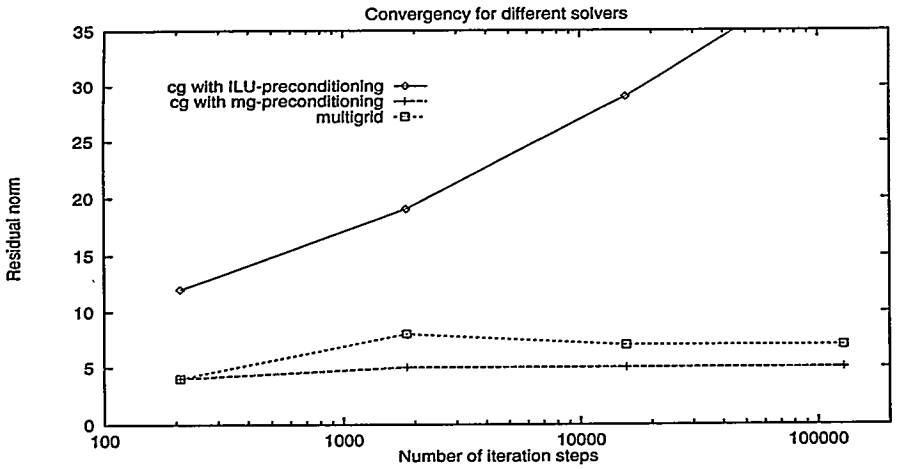


FIG. 3. *Number of iteration steps, different levels,$u_3(x,y,z)$*

Convergence of multigrid and cg with multigrid preconditioning is independent of the number of unknowns which nicely corresponds to theorem 2.1, while ILU-preconditioned cg shows significant growth in the number of iteration steps. It has to be noted though that one iteration step of cg with ILU-preconditioning is much "cheaper" than one step of cg with multigrid-preconditioning or of multigrid, resp. Interpretation of the results shown in Fig. 1-3 has to keep this in mind.

REFERENCES

[1] T. ARBOGAST AND Z. CHEN, *On the implementation of mixed methods as nonconforming methods for second-order elliptic problems*, Math. Comput., 64 (1995), pp. 943–972.

[2] D. N. ARNOLD AND F. BREZZI, *Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates*, M^2AN Math. Modelling and Numer. Anal., 19 (1985), pp. 7–32.

[3] I. BABUŠKA, M. GRIEBEL, AND J. PITKARÄNTA, *The problem of selecting the shape functions for a p-type finite element*, TR 88-36, Institute for Physical Science and Technology, University of Maryland, College Park, November 1988.

[4] F. BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, Springer, New York Berlin Heidelberg Tokyo, 1991.

[5] J. P. HENNART, *A general family of nodal schemes*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 264–287.

[6] J. P. HENNART AND E. DEL VALLE, *On the relationship between nodal schemes and mixed-hybrid finite elements*, Numer. Methods Partial Differential Equations, 9 (1993), pp. 411–430.

[7] J. C. NÉDÉLEC, *Mixed finite elements in $\mathbb{R}^3$*, Numer. Math., 35 (1980), pp. 315–341.

[8] ——, *A new family of mixed finite elements in $\mathbb{R}^3$*, Numer. Math., 50 (1986), pp. 57–81.

[9] W. SCHMID, *Numerische Behandlung der Neutronendiffusion auf adaptiven nichtuniformen Gittern*, PhD thesis, Math.-Nat. Fakultät der Universität Augsburg, Augsburg, 1996. In preparation.

A Unidirectional Approach for d-Dimensional Finite Element Methods of Higher Order on Sparse Grids

— Extended Abstract —

Hans-Joachim Bungartz
Institut für Informatik
Technische Universität München
D-80290 München, Germany
e-mail: bungartz@informatik.tu-muenchen.de

1 Introduction

In the last years, sparse grids have turned out to be a very interesting approach for the efficient iterative numerical solution of elliptic boundary value problems [2, 3, 5, 6, 8]. In comparison to standard (full grid) discretization schemes, the number of grid points can be reduced significantly from $O(N^d)$ to $O(N(\log_2(N))^{d-1})$ in the d-dimensional case, whereas the accuracy of the approximation to the finite element solution is only slightly deteriorated: For piecewise d-linear basis functions, e. g., an accuracy of the order $O(N^{-2}(\log_2(N))^{d-1})$ with respect to the L_2-norm and of the order $O(N^{-1})$ with respect to the energy norm has been shown. Furthermore, regular sparse grids can be extended in a very simple and natural manner to adaptive ones, which makes the hierarchical sparse grid concept applicable to problems that require adaptive grid refinement, too.

Starting from d-dimensional basis functions that are created from the one-dimensional ones by a tensor product approach, sparse grids allow the formulation of unidirectional algorithms for partial differential equations which are essentially independent of the number d of dimensions. Those unidirectional techniques are advantageous, since most of the algorithmic development can be done in the (simple) one-dimensional situation, whereas the generalization to $d > 1$ just results in additional outer loops or further levels of recursion.

Furthermore, it has been shown in [4] that sparse grids allow an accuracy of the finite element solution of $O(M^{-p})$ with respect to the energy norm, where M denotes the total number of unknowns involved in the given d-dimensional problem, and p is the polynomial degree of the basis functions used. Because of the accuracy's independence of the number d of dimensions, the sparse grid concept looks quite promising for the construction of efficient high order techniques in three or more dimensions.

In this paper, a unidirectional sparse grid Poisson solver for an arbitrary number d of dimensions and for various polynomial degrees p of the finite element basis functions is presented, together with first numerical examples. Combining the unidirectional sparse grid concept with higher order finite elements, this algorithm can be seen as a step on the way to the implementation of h-p-version-type finite element methods for arbitrary d on sparse grids.

2 The Problem

We want to develop a new technique for the efficient numerical solution of elliptic partial differential equations. In this paper, the approach is presented for the Laplacian on a unit domain. Concerning the discretization, finite elements, hierarchical tensor product bases of piecewise arbitrary polynomial degree, and sparse grids are used. The crucial part of the method is an algorithm that multiplies a matrix A (the stiffness matrix) with a vector u (the actual solution or an increment to it). Such a kernel allows a large flexibility with respect to the solver that is to be used later (cg, multi-level-techniques, even the integration in domain decomposition or recursive substructuring; at the moment, a diagonally preconditioned cg-iteration is used). The algorithm works in a unidirectional way and can handle an arbitrary number of dimensions by means of a recursive reduction of the general d-dimensional situation to the one-dimensional case. Thus, most of the algorithmic development can be done in a simpler one-dimensional context.

3 The Sparse Grid Concept

The use of hierarchical bases for finite element discretizations as proposed by Yserentant [7] and others instead of standard nodal bases stood at the beginning of the sparse grid idea, together with a tensor-product-type approach for the generalization from the one-dimensional to the d-dimensional case. For the corresponding subspace splitting of a full grid discretization space in two dimensions with piecewise bilinear hierarchical basis functions as in the left part of figure 1, it can be seen that the dimension (i. e., the number of grid points) of all subspaces with $i_1 + i_2 = c$ is 2^{c-2}. Furthermore, it has been shown in [3] that the contribution of all those subspaces with $i_1 + i_2 = c$ to the interpolant of a function u is of the same order $O(2^{-2c})$ with respect to the L_2- or maximum norm and $O(2^{-c})$ with regard to the energy norm, if u fulfills the smoothness requirement $\frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} \in \mathcal{C}^0(\bar{\Omega})$ for the two-dimensional and $\frac{\partial^{2d} u}{\partial x_1^2 \ldots \partial x_d^2} \in \mathcal{C}^0(\bar{\Omega})$ for the general d-dimensional case, respectively. Here, Ω denotes the underlying domain. Therefore, it turns out to be more reasonable to deal with a triangular subspace scheme as given in the right part of figure 1. This leads us to the so-called *sparse grids*. For a formal definition of sparse grids, see [2, 3, 8].

Besides the regular sparse grids that result from skipping certain subspaces according to figure 1, adaptive grid refinement can be realized in the sparse grid context in a very straightforward way. Since we use recursive dynamic data structures like binary trees for the implementation, and since the value of a hierarchical basis function, the hierarchical surplus, can be used itself to indicate the smoothness of u at the corresponding grid point and, consequently, the necessity to refine the grid here, no additional work has to be done to implement adaptive refinement. Figure 2 shows a two-dimensional regular sparse grid and a three-dimensional adaptive one with singularities at the re-entrant corner and along the edges.

Speaking about the most important properties of sparse grids, we at least have to look at the number of grid points involved and at the approximation accuracy of piecewise d-linear hierarchical basis functions on sparse grids. For a detailed analysis, we once again refer to [3, 8]. For a d-dimensional problem, the approach described above and illustrated in figure 1 leads to regular sparse grids with $O(N(\log_2(N))^{d-1})$ grid points, if N denotes the number of grid points in one dimension (i. e., $\frac{1}{N}$ is the smallest mesh width occurring). A variant even leads to regular sparse grids with $O(N)$ grid points. These results have to be compared with the $O(N^d)$ points of regular full grids. Concerning the approximation quality, the accuracy of the sparse grid interpolant is only slightly deteriorated from $O(N^{-2})$ to $O(N^{-2}(\log_2(N))^{d-1})$ with respect

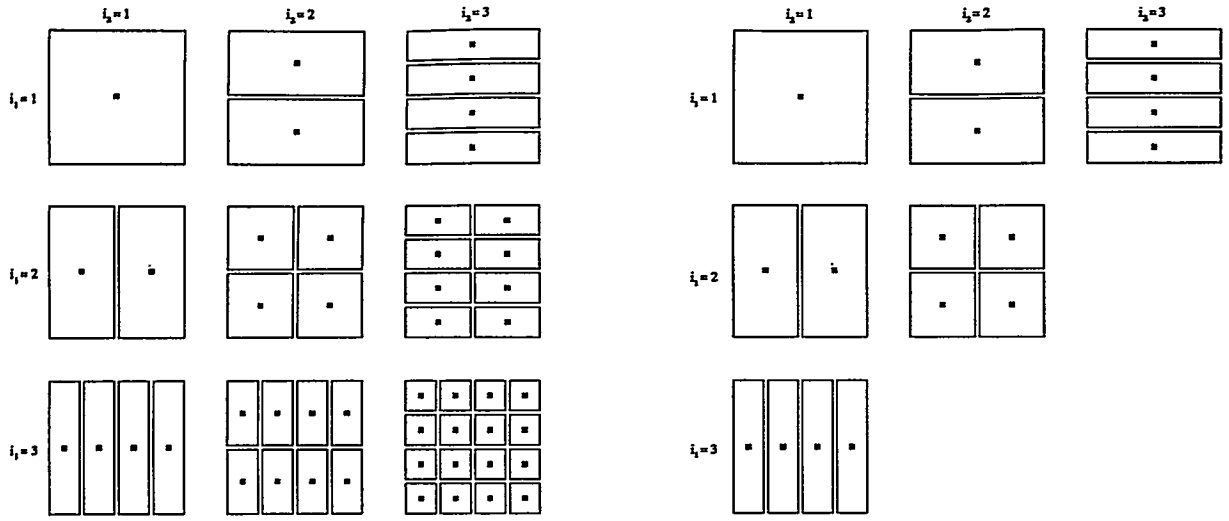


Figure 1: Subspace splitting of a full grid (left) and a sparse grid space (right).

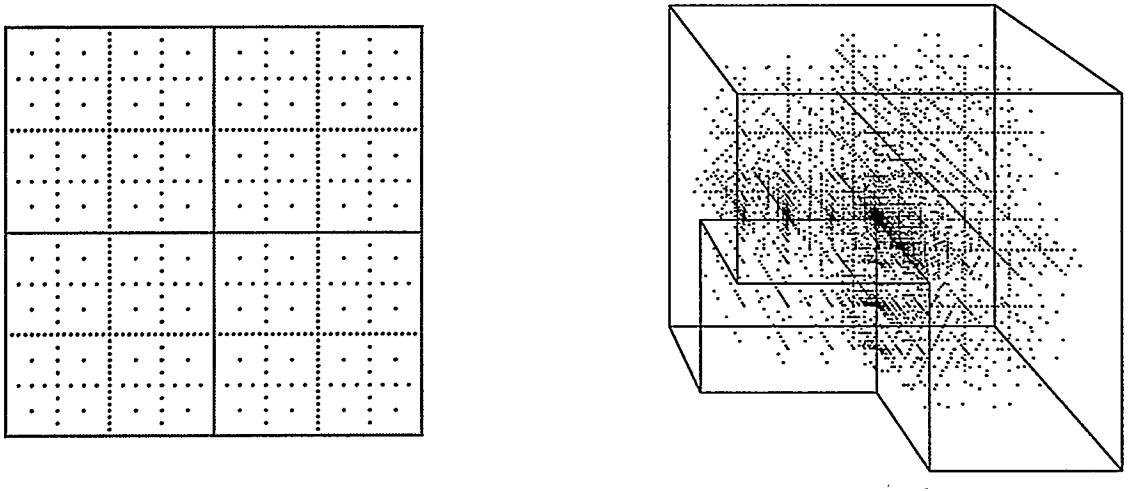


Figure 2: A regular and an adaptive sparse grid.

to the L_2- or maximum norm. With regard to the energy norm, both the sparse grid interpolant and the finite element approximation to the solution of the given boundary value problem stay of the order $O(N^{-1})$.

Thus, sparse grids enable us to gain a factor of 2 in accuracy for arbitrary number d of dimensions by just doubling the number of grid points. Since the smoothness requirements can be overcome by adaptive grid refinement, sparse grids are a very efficient approach for the solution of partial differential equations.

Recently, the class of problems that can be treated with sparse grid methods has been significantly extended. Pflaum developed a first algorithm for the treatment of general elliptic differential operators of second order in two dimensions, and Dornseifer developed a mapping technique to deal with curvilinear domains [6].

4 The Algorithmic Principle

For d-dimensional tensor-product-type hierarchical basis functions

$$\varphi_i(x_1, \ldots, x_d) := \prod_{l=1}^{d} \varphi_{i,l}(x_l) \tag{1}$$

with some kind of one-dimensional hierarchical basis function $\varphi_{i,l}(x_l), 1 \leq l \leq d$, an entry $a_{i,j}$ of the stiffness matrix A for the Laplacian is of the form

$$a_{i,j} = \sum_{k=1}^{d} \left(\int_{\Omega_{i,k} \cap \Omega_{j,k}} \frac{\partial \varphi_{i,k}(x_k)}{\partial x_k} \cdot \frac{\partial \varphi_{j,k}(x_k)}{\partial x_k} \, dx_k \cdot \prod_{l \neq k} \int_{\Omega_{i,l} \cap \Omega_{j,l}} \varphi_{i,l}(x_l) \cdot \varphi_{j,l}(x_l) \, dx_l \right), \tag{2}$$

where $\Omega_{i,k} = \mathrm{supp}(\varphi_{i,k}(x_k))$. For each k, the corresponding summand in (2) is calculated separately in an own pass through the data structure. Since all of those d terms are products of d one-dimensional integrals, the computation of the $a_{i,j}$ requires just two one-dimensional procedures, namely integration routines for

$$\int_{\Omega_{i,k} \cap \Omega_{j,k}} \frac{\partial \varphi_{i,k}(x_k)}{\partial x_k} \cdot \frac{\partial \varphi_{j,k}(x_k)}{\partial x_k} \, dx_k \tag{3}$$

and

$$\int_{\Omega_{i,k} \cap \Omega_{j,k}} \varphi_{i,k}(x_k) \cdot \varphi_{j,k}(x_k) \, dx_k. \tag{4}$$

Actually, for an efficient calculation of our matrix-vector-product Au, we do not need the $a_{i,j}$ themselves, but just for each unknown u_i the sum $\sum_{j=1}^{N} a_{i,j} u_j$. This is done in a recursive way, such that we get all of those sums during a single pass through the data structure. Roughly spoken, we start with a vector u consisting of the actual solution u_i in all grid points i and make a copy uu of it. Then, with u, a top-down-pass (called *down* in the following) through the data structure in hierarchical order is done, and with uu, we make a bottom-up pass (called *up*). After that, u_i contains the sums of all values $a_{i,j} u_j$ originating from grid points j hierarchically higher than i and from i itself, and uu_i contains all $a_{i,j} u_j$ from grid points or unknowns, resp., hierarchically lower than i. Finally, $u + uu$ provides $\sum_{j=1}^{N} a_{i,j} u_j$ in each grid point i. This one-dimensional algorithmic scheme is shown in figure 3.

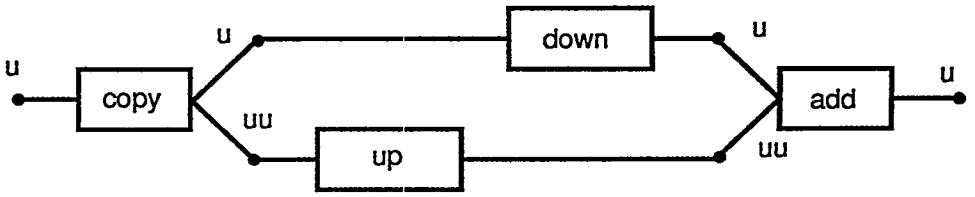


Figure 3: Scheme of the one-dimensional algorithm.

Note that, in the one-dimensional case (i. e. for $u'' = f$), only integrals of type (3) occur.

The recursive extension of the algorithmic principle indicated in figure 3 to the general d-dimensional case is given by figure 4. There, for $d = 2$, e. g., the grey boxes entitled *rec_ext* have to be replaced by the one-dimensional scheme from figure 3. Thus, due to the copy-process, $2d$ variables are necessary per grid point.

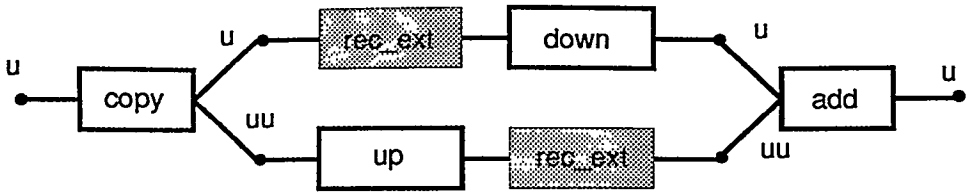


Figure 4: Scheme of the one-dimensional algorithm.

5 The Polynomial Bases

Next, we have to choose a suitable hierarchical basis. Due to the tensor product approach, we only have to deal with the one-dimensional case. Each basis function $\varphi_{i,k}$ is defined by three natural conditions: its value is 1 in grid point i and 0 at the two boundary points of $\Omega_{i,k}$. For a polynomial degree $p > 2$, this is not sufficient for fixing the basis functions. Therefore, we add additional interpolation conditions in grid points *outside* $\Omega_{i,k}$, i. e., in hierarchical ancestors of i. Thus, depending on the position of i in the grid, we get two different types of basis functions for $p = 3$, four types for $p = 4$, eight for $p = 5$, and so on. Figures 5 and 6 show the quadratic, cubic, and quartic basis functions.

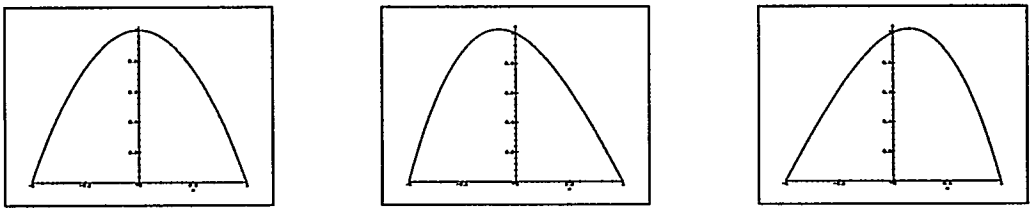


Figure 5: Hierarchical basis functions for $p = 2$ (left) and $p = 3$ (centre and right).

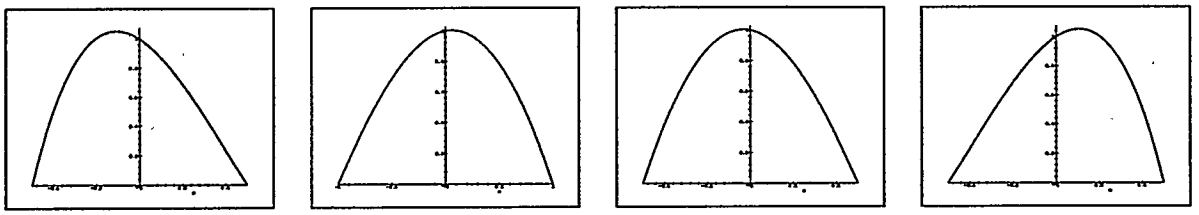


Figure 6: Hierarchical basis functions for $p = 4$.

If we number the types of basis functions in their natural order, i. e., the quadratic type gets number 0, the cubic types get numbers 1 and 2, the quartic ones numbers 3 through 6, and so on, we can indicate the actual type of basis function at each grid point (see figure 7 for a possible choice if $p = 4$).

Each basis function is characterized by the vector of its Taylor-coefficients $a_l, 0 \leq l \leq p_{max}$, i. e.

$$\varphi_{i,k}(x_k) = \sum_{l=0}^{p_{max}} a_l \frac{x_k^l}{l!}, \tag{5}$$

where the maximal degree p_{max} has to be specified at the beginning. Furthermore, the local interpolant or the local approximation to the solution, resp., can be represented as such a vector

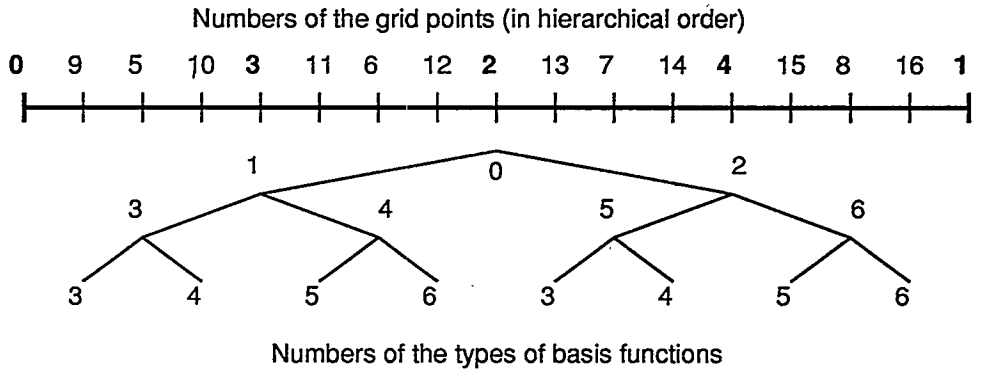


Figure 7: Basis functions and grid points.

of Taylor-coefficients, too. This allows a very elegant and simple access to p-adaptivity, since the actual local degree can be varied by taking another a_l into account or by omitting it, resp.

Now, let's remember the down- and up-procedures in figure 3 and 4. During the top-down-pass, we have to pass on the information concerning the local interpolant from the father to its two sons. It can be shown that this is just a multiplication of an upper triangular $(p+1)\times(p+1)$ Toeplitz-matrix T constant for all grid points with the local vector of Taylor-coefficients and some additional scaling. Furthermore, the process of passing on the information of the accumulated $a_{i,j}u_j$ from the sons to their father during the bottom-up-pass can be shown to be equivalent to a matrix-vector-multiplication with T^T and some scaling. This fact shows the close relation of the down- and up-procedures.

6 The Implementation and Conclusions

Up to now, all programming has been done in a rapid prototyping spirit: All code necessary for the construction of the different types of basis functions of arbitrary polynomial degree p, for the calculation of their Taylor-coefficients, and for the computation of certain type-dependent integrals has been written in Maple. All of those calculations can be done in a kind of setup process, and the Maple output can be directly used as input for the actual solver. Overall, for a maximal polynomial degree p, about $(p+1)\cdot 2^{p-1}$ coefficients and (simple) integrals have to be pre-computed and stored.

The code for the algorithm itself has been written in Perl. Originally developed for purposes of system administration, Perl has turned out to be a very efficient and powerful tool for prototyping in a numerical environment, too, since it combines the functionality of standard programming languages with the simple and direct programming possibilities of a shell script (interpreter) language. Especially, in comparison with C or C++, all of the declaration overhead can be avoided, and the development and tests of code can be significantly accelerated since there is no compiling. Furthermore, Perl has some interesting features like associative arrays which allow a much easier programming of adaptive hierarchical structures than pointers do. In this implementation, e. g., each grid point in the adaptive sparse grid is characterized by a hierarchical number in each dimension (see figure 7). In a three-dimensional problem, e. g., the centre of the cube has the number 2 in each dimension. This is interpreted as a string "2 2 2", which is now used as the hash key for the associative array. Thereby, the programmer always has a direct access to the entries in the array, without having to organize, maintain, and follow wide nets of references.

Of course, from the point of view of run-time efficiency, an interpreter language like Perl cannot keep up the pace with standard programming languages. Therefore, once the algorithmic idea has been verified and has turned out to be quite promising, an implementation in a language like C or C++ has to follow, which will be the future work in this area.

As a short summary of this extended abstract, let us recall the most important properties of this new sparse grid algorithm. First of all, it allows to deal with an arbitrary number of dimensions just by changing the input parameter d. Second, it can handle piecewise polynomials of an arbitrary degree p, which – together with the sparse grid efficiency – leads to a very high accuracy [4]. Third, the algorithmic structure is totally independent of p. All information concerning the different types of basis functions is pre-computed and stored in a table, and during the iteration, these tables are used depending on the local type of basis function. Finally, the storage requirement per grid point is independent of p. The grid points only hold the value of the function and other information necessary for the cg-iteration, but everything that depends in size on p is organized and stored on the stack, but not along with the data structure.

References

[1] I. Babuška and M. Suri, *The p- and h-p-versions of the finite element method: An overview*, Comput. Methods Appl. Mech. Engrg., 80 (1990), pp. 5–26.

[2] H.-J. Bungartz, *An adaptive Poisson solver using hierarchical bases and sparse grids*, in Iterative Methods in Linear Algebra, P. de Groen and R. Beauwens, eds., Elsevier, Amsterdam, 1992, pp. 293–310.

[3] H.-J. Bungartz, *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung*, Dissertation, Institut für Informatik, TU München, 1992.

[4] H.-J. Bungartz, *Concepts for higher order finite elements on sparse grids*, in Proceedings of the the 3^{rd} Int. Conf. on Spectral and High Order Methods, June 1995, Houston, L. R. Scott, ed., Houston Journal of Mathematics, to appear.

[5] H.-J. Bungartz, M. Griebel, and U. Rüde, *Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems*, Comput. Methods Appl. Mech. Engrg., 116 (1994), pp. 243–252.

[6] Th. Dornseifer and C. Pflaum, *Discretization of elliptic differential equations on curvilinear bounded domains with sparse grids*, to appear in Computing.

[7] H. Yserentant, *On the multilevel splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

[8] C. Zenger, *Sparse grids*, in Parallel Algorithms for Partial Differential Equations: Proceedings of the 6^{th} GAMM-Seminar, Kiel, January 1990, Notes on Numerical Fluid Mechanics 31, W. Hackbusch, ed., Vieweg, Braunschweig, 1991.

TWO–LEVEL METHOD WITH COARSE SPACE SIZE INDEPENDENT CONVERGENCE *

PETR VANĚK , RADEK TEZAUR , MARIAN BREZINA † AND JITKA KŘÍŽKOVÁ‡

1. INTRODUCTION. The basic disadvantage of the standard two-level method is the strong dependence of its convergence rate on the size of the coarse-level problem. In order to obtain the optimal convergence result, one is limited to using a coarse space which is only a few times smaller than the size of the fine-level one. Consequently, the asymptotic cost of the resulting method is the same as in the case of using a coarse-level solver for the original problem. Today's two-level domain decomposition methods typically offer an improvement by yielding a rate of convergence which depends on the ratio of fine and coarse level only polylogarithmically ([1], [2], [3], [5], [4], [6]). However, these methods require the use of local subdomain solvers for which straightforward application of iterative methods is problematic, while the usual application of direct solvers is expensive.

We suggest a method diminishing significantly these difficulties. Following the unpublished technical report [8], we develop a simple abstract framework based on the concept of smoothed aggregation introduced in [9] with aggregates derived from the system of nonoverlapping subdomains. We show that the smoothing of the coarse-space by an appropriate polynomial of degree about N_{es}^{-d} (symbol d denotes the dimension of the problem to be solved and N_{es} is the characteristic number of degrees of freedom per subdomain) can assure the coarse-space size independent convergence. The associated cost is significantly smaller than that of the local solvers in the case of standard domain decomposition. Moreover, it decreases as d increases.

Because of the page limit, we only apply the abstract framework for the case of scalar equation with jumps in coefficients. More general problems and numerical experiments will be treated in [7].

2. ABSTRACT FRAMEWORK. We are interested in a numerical solution of a system of linear algebraic equations

$$A\mathbf{x} = \mathbf{b} \tag{1}$$

with a symmetric positive definite $n \times n$ matrix A. Let $P : \mathbb{R}^m \to \mathbb{R}^n$, $m << n$ be a linear injective tentative prolongator and $S \in [\mathbb{R}^n]$ a symmetric smoother commuting with A. Let us set

$$A_S = S^2 A, \quad S' = I - \frac{\omega}{\rho} A_S, \quad \omega \in (0,2), \quad \rho = \rho(A_S). \tag{2}$$

* This research was supported by NSF grants ASC-9121431 and ASC-9217394.
This paper has been submitted for publication elsewhere.

† Center for Computational Mathematics, University of Colorado at Denver, Denver, CO 80217-3364.
‡ UWB, Americká 42, 30614 Plzeň, Czech Republic

Furthermore, let $\mathbf{x} \leftarrow \mathcal{S}_S(\mathbf{x}, \mathbf{b})$ and $\mathbf{x} \leftarrow \mathcal{S}_{S'}(\mathbf{x}, \mathbf{b})$ be relaxation methods consistent with (1) such that their linear parts are matrices S and S'. Our algorithm is a standard variational two-level method with a smoothed prolongator SP, a pre-smoother $\mathcal{S}_S$ and a post-smoother $\mathcal{S}_{S'}$.

ALGORITHM 2.1. *Given the initial approximation* $\mathbf{x}$,

repeat

1. $\mathbf{x} \leftarrow \mathcal{S}_S(\mathbf{x}, \mathbf{b})$,
2. *solve* $(P^T A_S P)\mathbf{v} = P^T S(A\mathbf{x} - \mathbf{b})$,
3. $\mathbf{x} \leftarrow \mathbf{x} - SP\mathbf{v}$,
4. $\mathbf{x} \leftarrow \mathcal{S}_{S'}(\mathbf{x}, \mathbf{b})$

until convergence;

5. *Post process* $\mathbf{x} \leftarrow \mathcal{S}_S(\mathbf{x}, \mathbf{b})$.

In the following, we will prove that, for a suitable S and P, steps 1–4 of the algorithm ensure convergence independent of the dimension ratio n/m in the A_S—norm. The postprocessing step 5 of the algorithm enables us to prove the same result in the energy norm of the original problem (1). The main disadvantage of the convergence estimate in A_S—norm is its indirect coarse-space dependence as for a smaller coarse-space we need a more powerfull smoother S to get the optimal convergence result.

ASSUMPTION 2.2. *Let the smoother* S *be a symmetric matrix that commutes with* A, *and* $\rho(S) \leq 1$. *We assume that the tentative prolongator* P *satisfies the weak approximation property in the following form: For every* $\mathbf{u} \in \mathbb{R}^n$, *there exists* $\mathbf{v} \in \mathbb{R}^m$ *such that*

$$\|\mathbf{u} - P\mathbf{v}\| \leq C_1 C_D(m,n)\rho^{-1/2}(A)\|\mathbf{u}\|_A. \tag{3}$$

For the prolongator smoother S, *we require*

$$\rho(S^2 A) \leq C_2^2 C_D^{-2}(m,n)\rho(A), \tag{4}$$

where $C_D(m,n), C_1, C_2 > 0$, *and* C_1, C_2 *do not depend on* m *and* n.

REMARK 2.3. *For second order problems, we typically have* $C_D(m,n) = H/h$ *(the ratio of local meshsizes on the coarse and the fine level). In Section 3 we construct* S *as a suitable polynomial in* A *for which* $\rho(S^2 A) \approx N^{-2}\rho(A)$, *where* N *denotes the degree of* S. *In order to satisfy Assumption 2.2, we need* $N \approx H/h$. *This choice yields a coarse level matrix* $(SP)^T A(SP)$ *with a number of nonzero entries per row uniformly bounded with respect to* H/h. *Detailed arguments will be given in Section 3.*

The following theorem shows that, under Assumption 2.2, the convergence rate of Algorithm 2.1 is independent of dimensions m and n of the coarse and fine spaces.

THEOREM 2.4. *Let* $\mathbf{e}_i$ *denote the error after* i *iterations given by steps 1–4 of Algorithm 2.1 , and* $\mathbf{e}_i^S = S\mathbf{e}_0$ *the error smoothed by step 5. Then, it holds that*

$$\|\mathbf{e}_{i+1}\|_{A_S}^2 \leq (1 - C_3)\|\mathbf{e}_i\|_{A_S}^2, \quad \textit{and} \quad \|\mathbf{e}_i^S\|_A^2 \leq (1 - C_3)^i \|\mathbf{e}_0\|_A^2, \tag{5}$$

where $C_3 = \frac{(C_1 C_2)^{-2}\omega(2-\omega)}{1+(C_1 C_2)^{-2}\omega(2-\omega)} > 0$. *Here* ω *is the damping parameter from* (2).

Proof. Since $\rho(S) \leq 1$ and $\mathbf{e}_i^S = S\mathbf{e}_i$, we have $\|\mathbf{e}_i^S\|_A = \|\mathbf{e}_i\|_{A_S}$ and $\|\mathbf{e}_0\|_{A_S} \leq \|\mathbf{e}_0\|_A$. Therefore, the second inequality in (5) follows from the first one.

It is obvious that the linear part of the steps 1–4 is given by

$$S'[I - SP(P^T A_S P)^{-1} P^T SA]S = S'S[I - P(P^T A_S P)^{-1} P^T A_S].$$

Thus, the method can be viewed as a standard two-level method for solving a problem with matrix A_S (in place of A) and prolongator P (in place of SP).

Since $I - P(P^T A_S P)^{-1} P^T A_S$ is the A_S–orthogonal projection onto A_S–orthogonal complement of Range (P) (i.e., projection onto Ker $(P^T A_S)$), $S'S = SS'$, $\rho(S) \leq 1$, and $\rho(S') \leq 1$, we have

$$\|S'S[I - P(P^T A_S P)^{-1} P^T A_S]\|_{A_S}^2 \leq \sup_{\mathbf{x} \in \mathrm{Ker}\,(P^T A_S)} \min\left\{ \frac{\|S\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2}, \frac{\|S'\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2} \right\}. \quad (6)$$

In the rest of the proof, we will show that at least one of the expressions in the minimum above is bounded by $1 - C_3$ for any $\mathbf{x} \in$ Ker $(P^T A_S)$.

We first express $\|S'\mathbf{x}\|_{A_S}/\|\mathbf{x}\|_{A_S}$ in terms of $\|S\mathbf{x}\|_{A_S}/\|\mathbf{x}\|_{A_S}$. It is easy to see that

$$\frac{\|A_S\mathbf{x}\|^2}{\|\mathbf{x}\|_{A_S}^2} \geq C_E\rho(A_S) \quad \text{implies} \quad \frac{\|S'\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2} \leq 1 - C_E\omega(2-\omega). \quad (7)$$

Let us recall that $A_S = AS^2$, and A and S commute. Hence, we can write $\|S^2\mathbf{x}\|_A^2 = \|S\mathbf{x}\|_{A_S}^2$, and

$$\frac{\|A_S\mathbf{x}\|^2}{\|\mathbf{x}\|_{A_S}^2} = \frac{\|A_S\mathbf{x}\|^2}{\|S^2\mathbf{x}\|_A^2} \frac{\|S^2\mathbf{x}\|_A^2}{\|\mathbf{x}\|_{A_S}^2} = \frac{\|AS^2\mathbf{x}\|^2}{\|S^2\mathbf{x}\|_A^2} \frac{\|S\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2}. \quad (8)$$

Now, consider $\mathbf{x} \in$ Ker $(P^T A_S) =$ Ker $(P^T AS^2)$. Then, setting $\mathbf{u} = S^2\mathbf{x}$, we have $\mathbf{u} \in$ Ker $(P^T A) =$ Range $(P)^{\perp_A}$. From the weak approximation condition (3), we estimate the ratio $\frac{\|AS^2\mathbf{x}\|^2}{\|S^2\mathbf{x}\|_A^2}$ using the standard orthogonality argument: For $\mathbf{v} \in \mathbb{R}^m$ from (3), we obtain

$$\|\mathbf{u}\|_A^2 = (A\mathbf{u}, \mathbf{u}) = (A\mathbf{u}, \mathbf{u} - P\mathbf{v}) \leq \|A\mathbf{u}\| \, \|\mathbf{u} - P\mathbf{v}\| \leq C_1 C_D(m,n)\rho^{1/2}(A)\|A\mathbf{u}\| \, \|\mathbf{u}\|_A.$$

Therefore, $\|A\mathbf{u}\| \geq C_1^{-1} C_D^{-1}(m,n)\rho^{1/2}(A)\|\mathbf{u}\|_A$. Substituting this estimate into (8) and using Assumption 2.2, we get

$$\frac{\|A_S\mathbf{x}\|^2}{\|\mathbf{x}\|_{A_S}^2} \geq C_1^{-2} C_D^{-2}(m,n)\rho(A) \frac{\|S\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2} \geq (C_1 C_2)^{-2}\rho(A_S) \frac{\|S\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2}. \quad (9)$$

Thus, by (7)

$$\frac{\|S'\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2} \leq [1 - \frac{\|S\mathbf{x}\|_{A_S}^2}{\|\mathbf{x}\|_{A_S}^2} (C_1 C_2)^{-2}\omega(2-\omega)].$$

Since $\|S\mathbf{x}\|_{A_S}^2/\|\mathbf{x}\|_{A_S}^2 \leq 1$, we may finally write using (6)

$$\|S'S[I - P(P^T A_S P)^{-1} P^T A_S]\|_{A_S}^2 \leq \sup_{\alpha \in [0,1]} \min\left\{\alpha, 1 - \alpha(C_1 C_2)^{-2}\omega(2-\omega)\right\}.$$

The expression on the right hand side is bounded by $\frac{1}{1+(C_1 C_2)^{-2}\omega(2-\omega)}$ which completes the proof. □

3. EXAMPLE OF A TENTATIVE COARSE SPACE AND PROLONGATOR SMOOTHER. Let $\Omega \subset \mathbb{R}^2$ be a Lipschitz domain and $\mathcal{T}$ be a shape-regular (locally quasiuniform) finite element mesh on Ω. Let $V_{\mathcal{T}}$ be $P1$ or $Q1$ finite element space associated with the mesh $\mathcal{T}$ with zero Dirichlet boundary conditions imposed at some nodes of $\mathcal{T} \cap \partial\Omega$. Note that, for the purpose of numerical solution of the discretized problem, we do not need any assumptions on the form or measure of the part of the boundary with Dirichlet conditions imposed. For simplicity, we assume that the finite element basis functions φ_i are scaled so that $\|\varphi_i\|_{L^\infty} = 1$. We consider the following elliptic model problem: Find $u \in V_{\mathcal{T}}$ such that

$$A_\Omega(u,v) = (f,v)_{L^2(\Omega)} \quad \forall v \in V_{\mathcal{T}}, \quad A_\Omega(u,v) = \sum_{i=1}^{2} \int_\Omega a(x) \frac{\partial u(x)}{\partial x_i} \frac{\partial v(x)}{\partial x_i} dx. \tag{10}$$

We allow large variation of $a(x)$ between the subdomains; more detailed assumptions on $a(x)$ will be specified below.

Finite element discretization of (10) on $V_{\mathcal{T}}$ leads to a system of linear algebraic equations with a symmetric positive definite matrix $A_{\mathcal{T}}$. In order to accomodate discontinuities of the coefficient $a(x)$, we will solve the system with a diagonally scaled matrix $A = D^{-1/2} A_{\mathcal{T}} D^{-1/2}$ instead, where $D = \mathrm{diag}(A_{\mathcal{T}})$.

For the sake of construction of the tentative prolongator P we need a system $\{\Omega_i\}_{i=1}^m$ of closed disjoint subdomains of Ω such that each subdomain Ω_i is a simply connected closure of an aggregate of elements. We assume that each node of the underlying finite element mesh belongs to exactly one of the subdomains and that there is a layer one element wide between two neighboring subdomains. Further, we assume that the family of subdomains $\{\Omega_i\}$ satisfies the following properties:

ASSUMPTION 3.1.

(i) We assume that there is about the same number of elements in each subdomain Ω_i. Let us denote the characteristic number of elements per subdomain by N_{es} and set $\hat{h} = N_{es}^{-1/2}$. We require that subdomain Ω_i can be mapped onto a reference subdomain $\hat{\Omega} = [0,1] \times [0,1]$ by a one-to-one locally Lipschitz mapping G_i:

$$||\partial G_i(x)|| \le C \frac{\hat{h}}{h(x)}, \quad ||\partial G_i^{-1}(\hat{x})|| \le \frac{h(x)}{c\hat{h}}, \quad \forall \hat{x} = G_i(x), x \in \Omega_i, \tag{11}$$

where $h(x)$ is the local meshsize in the neighborhood of x and $c, C > 0$ are constants uniform with respect to i. Symbol $||\cdot||$ denotes a matrix operator norm.

(ii) The coefficient $a(x)$ is allowed only a modest variation within each subdomain in the sense that $a(x) \approx a_i > 0, \quad \forall x \in \Omega_i$.

(iii) If Ω_i and Ω_j are adjacent subdomains (there exists $T \in \mathcal{T}$ so that $\partial T \cap \partial\Omega_i \neq \emptyset$ and $\partial T \cap \partial\Omega_j \neq \emptyset$) and $a_i >> a_j$, the jump in $a(x)$ occurs along $\partial\Omega_i$. In other words, the discontinuity is located on the boundary of the subdomain with the larger value of $a(x)$.

Conditions (11) imply that an element $T \subset \Omega_i$ of size $h(x)$ is mapped by G_i onto $G_i(T)$ of size about $\hat{h}$. Thus, G_i maps locally quasi-uniform mesh on Ω_i onto a quasi-uniform mesh of meshize $\hat{h}$, and subdomains Ω_i are reasonable aggregates consisting of

about $N_{es} = \hat{h}^{-2}$ elements. If the mesh $\mathcal{T}$ is quasi-uniform ($h(x) \approx h$), then $h(x)/\hat{h}$ can be viewed as the characteristic size of Ω_i. Note that, if a subdomain decomposing algorithm uses only the adjacency of elements or nodes and generates shape regular subdomains in the case of quasiuniform mesh, then for locally quasi-uniform meshes it can be expected to generate subdomains satisfying (11).

The purpose of assumption *(iii)* is to ensure that the basis function φ_j associated with a node $v_j \in \Omega_i$, satisfies

$$a(\varphi_j, \varphi_j) \approx a_i. \tag{12}$$

If *(iii)* were not satisfied, (12) could be violated for the basis functions corresponding to the nodes on $\partial\Omega_j$ adjacent to a subdomain Ω_i with $a_i >> a_j$.

The tentative prolongator based on scaled aggregations is defined as follows.

ALGORITHM 3.2.

(i) Set $P_{ij} = \begin{cases} 1, & \text{if the node } v_i \text{ belongs to subdomain } \Omega_i, \\ 0, & \text{otherwise.} \end{cases}$

(ii) Set $P \leftarrow D^{1/2}P$, *where* $D = \mathrm{diag}(A_{\mathcal{T}})$.

For each subdomain, we introduce an index set F_i of all unconstrained (with no Dirichlet boundary conditions imposed) degrees of freedom associated with Ω_i. Let $\Pi : \mathbb{R}^n \to V_\tau$ denotes the finite element interpolator given by $\Pi \mathbf{x} = \sum_{j=1}^n x_j \varphi_j$, the local interpolator $\Pi_i \mathbf{x} = \sum_{j \in F_i} x_j \varphi_j$, and discrete $l^2(F_i)$−norm $\|\mathbf{x}\|^2_{l^2(F_i)} = \sum_{j \in F_i} x_j^2, \mathbf{x} \in \mathbb{R}^n$.

Let Ω_i' be a subset of Ω_i consisting of all elements $T \subset \Omega_i$ such that all degrees of freedom on T are unconstrained. On each subdomain we define a linear mapping $Q_i : \mathbb{R}^n \to \mathbb{R}^n$ (acting on the degrees of freedom of F_i only) to be the $l^2(F_i)$-orthogonal projection onto the one-dimesional space of vectors spanned by $\mathbf{c} \in \mathbb{R}^n$ such that $c_j = 1$ for $j \in F_i$, zero elsewhere.

LEMMA 3.3 (DISCRETE SCALED POINCARÉ-FRIEDRICHS INEQUALITY). *For every* $\mathbf{u} \in \mathbb{R}^n$, *it holds that*

$$\|\mathbf{u} - Q_i \mathbf{u}\|_{l^2(F_i)} \leq C N_{es}^{1/2} |\Pi_i \mathbf{u}|_{H^1(\Omega_i)}, \tag{13}$$

the constant $C > 0$ *depends on the constants from* (11) *and on the aspect ratios of elements in* Ω_i *only.*

Proof. Consider a transformed function $\hat{u} = u \circ G_i^{-1}$ i.e. $\hat{u}(\hat{x}) = u(G_i^{-1}(\hat{x})), \hat{x} \in \hat{\Omega}$. Let us define a weighted L^2−norm by $\|u\|_{L_h^2} = \|u(x)/h(x)\|_{L^2}$. Owing to (11), H^1−seminorm scales uniformly, i.e. $|u|_{H^1(\Omega_i)} \approx |\hat{u}|_{H^1(\hat{\Omega})}$. It can be easily seen that

$$\|u\|_{L_h^2(\Omega_i)} \approx \hat{h}^{-1} \|\hat{u}\|_{L^2(\hat{\Omega})} \quad \text{and} \quad \|\Pi_i \mathbf{x}\|_{L_h^2(\Omega')} \approx \|\mathbf{x}\|_{l^2(F_i)}.$$

Let $\mathbf{c} \in \mathbb{R}^n$ be the vector given by $c_j = 1$ for $j \in F_i$, zeroes elsewhere (as in the definition of Q_i). Then, by the equivalence of $l^2(F_i)$ and $L_h^2(\Omega_i')$ for finite element functions , $\Omega_i' \subset \Omega_i$, and the scaling above, we have for $\alpha \in \mathbb{R}$

$$\begin{aligned} \|\mathbf{u} - \alpha \mathbf{c}\|_{l^2(F_i)} &\approx \|\Pi_i \mathbf{u} - \alpha\|_{L_h^2(\Omega_i')} \\ \leq \|\Pi_i \mathbf{u} - \alpha\|_{L_h^2(\Omega_i)} &\approx \hat{h}^{-1} \|(\Pi_i \mathbf{u} - \alpha) \circ G_i^{-1}\|_{L^2(\hat{\Omega})}, \end{aligned} \tag{14}$$

Using the definition of Q_i, inequality (14), Poincaré-Friedrichs inequality on $\hat{\Omega}$ and the uniform scaling of H^1 seminorm, we obtain

$$\begin{aligned}\|\mathbf{u} - Q_i\mathbf{u}\|_{l^2(F_i)} &= \inf_{\alpha\in\mathbb{R}^1}\|\mathbf{u} - \alpha\mathbf{c}\|_{l^2(F_i)} \le C\hat{h}^{-1}\inf_{\alpha\in\mathbb{R}^1}\|(\Pi_i\mathbf{u})\circ G_i^{-1} - \alpha\|_{L^2(\hat{\Omega})} \\ &\le C\hat{h}^{-1}|(\Pi_i\mathbf{u})\circ G_i^{-1}|_{H^1(\hat{\Omega})} \le C\hat{h}^{-1}|\Pi_i\mathbf{u}|_{H^1(\Omega_i)},\end{aligned}$$

concluding the proof. □

Now we are ready to prove the weak approximation property (3).

LEMMA 3.4 (WEAK APPROXIMATION PROPERTY). *Under Assumption 3.1, the inequality* (3) *is satisfied with* $C_D(m,n) = N_{es}^{1/2}$ *and* C_1 *that depends only on constants from* (11) *and aspect ratios of elements.*

Proof. We set $Q = D^{1/2}(\sum_{i=1}^m Q_i)D^{-1/2}$. Let $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{x} = D^{1/2}\mathbf{u}$. Then, using (12),

$$\|\mathbf{u}\|_A^2 = \|\mathbf{x}\|_{A_\mathcal{T}}^2 = A_\Omega(\Pi\mathbf{x},\Pi\mathbf{x}) \ge \sum_{i=1}^m A_{\Omega_i}(\Pi_i\mathbf{x},\Pi_i\mathbf{x}) \ge C\sum_{i=1}^m a_i|\Pi_i\mathbf{x}|_{H^1(\Omega_i)},$$

$$\|\mathbf{u} - Q\mathbf{u}\|^2 = \|D^{1/2}(I - \sum_{i=1}^m Q_i)\mathbf{x}\|^2 \le C\sum_{i=1}^m a_i\|\mathbf{x} - Q_i\mathbf{x}\|_{l^2(F_i)}^2.$$

From here, setting $P\mathbf{v} = Q\mathbf{u}$ and using Lemma 3.3 and $\rho(A) \le C$, the statement follows. □

In the rest of this section we will discuss the choice of the prolongator smoother S. Let $\bar{\rho}$ be the estimate of $\rho(A)$ satisfying

$$\rho(A) \le \hat{\rho} \le C_\rho\rho(A). \tag{15}$$

For any integer $i \ge 0$, we define $\hat{\rho}_i = \frac{\hat{\rho}}{9^i}$, $A_0 = A$ and

$$S_i = \prod_{j=0}^{i-1} W_j, \quad W_j = I - \frac{4}{3}\hat{\rho}_j^{-1}A_j, \quad A_j = W_{j-1}^2A_{j-1}. \tag{16}$$

It is easy to see that $\deg(S_i) \le \frac{3}{2}3^i$. We choose the prolongator smoother $S = S_K$ for K such that

$$\deg(S_{K+1}) \ge qN_{es}^{1/2} \ge \deg(S_K), \tag{17}$$

where $q \in (0,1]$ is a given parameter.

THEOREM 3.5. *Let the tentative prolongator P be given by Algorithm 3.2 with the system of subdomains* $\{\Omega_i\}_{i=1}^m$ *satisfying Assumption 3.1. Let the prolongator smoother S be defined by* (15)–(17). *Then, the statement of Theorem 2.4 is valid with the constant C_3 independent of the meshsize, coefficients a_i, constant N_{es}, and boundary conditions. Moreover, the coarse-level matrix and the smoothed prolongator SP have a uniformly bounded number of nonzero entries per row.*

Proof. Due to Lemma 3.4, the approximation property (3) is satisfied with $C(m,n) = N_{es}^{1/2}$. Let us show that (4) holds with the same $C(m,n)$. From definition (16), we have

$S^2A = A_{K+1}$. By induction, we can prove $\rho(A_i) \leq \hat{\rho}_i$: For $i = 0$, the inequality holds by (15); assume it holds for $j \leq i$. Then, by (16)

$$\rho(A_{i+1}) = \max_{t \in \sigma(A_i)} (1 - \frac{4}{3}\hat{\rho}_i^{-1}t)^2 t \leq \max_{t \in [0,\hat{\rho}_i]} (1 - \frac{4}{3}\hat{\rho}_i^{-1}t)^2 t \leq \hat{\rho}_{i+1}.$$

Hence, $\rho(A_K) \leq 9^{-K}\hat{\rho}$. Considering that, by (17), $K \approx \log_3 N_{es}^{1/2}$, we get (4). The optimal convergence result now follows from Theorem 2.4.

Let us show that the number of nonzero entries per row of the coarse-level matrix $A_c = (SP)^T A(SP)$ is bounded uniformly with respect to N_{es}. It is easy to see that $[A_c]_{ij}$ can be nonzero only if $\mathrm{supp}(\Pi SP\mathbf{e}^i) \cap \mathrm{supp}(\Pi SP\mathbf{e}^j) \neq \emptyset$, where $\mathbf{e}^i$ is the i–th canonical basis vector of $\mathbb{R}^m$. Clearly, $\mathrm{supp}(\Pi P\mathbf{e}^i)$ is the domain Ω_i with one belt of surrounding elements added. Bounded overlaps of such supports are obvious. The smoother S adds at most qN_{es} strips of elements. Consequently, each support has a nonempty intersection with only a bounded number of other supports. □

THEOREM 3.6. *Let the assumptions of Theorem 3.5 be fulfilled and the Choleski factorization be used to solve the coarse-level problem. Then, the optimal number of elements per subdomain is $N_{es} \approx n^{2/5}$ and the system (1) can be solved to the level of truncation error in $O(n^{1.2})$ operations.*

Proof. We only consider the components of the algorithm which cost more than $O(n)$ operations. During the setup, such procedures involve evaluation of SP ($O(N_{es}^{1/2}n)$ operations) and Choleski factorization of the coarse-level matrix, which costs $O(m^2) = O(n^2/N_{es})$ operations. As Theorem 3.5 assures the optimal convergence result, we only have to perform $O(1)$ iterations. Nonscalable procedures during each iteration are the smoothing ($O(N_{es}^{1/2}n)$ operations) and the back substitution ($O(m^{1.5}) = O((n/N_{es}^{1/2})^{1.5})$. The statement follows by trivial manipulations. □

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), pp. 103–134.

[2] ——, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp., 53 (1989), pp. 1–24.

[3] M. DRYJA, B. F. SMITH, AND O. B. WIDLUND, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal., 31 (1994), pp. 1662–1694.

[4] C. FARHAT AND F. X. ROUX, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Int. J. Numer. Meth. Engng., 32 (1991).

[5] J. MANDEL, *Balancing domain decomposition*, Comm. in Numerical Methods in Engrg., 9 (1993), pp. 233–241.

[6] J. MANDEL AND R. TEZAUR, *Convergence of a substructuring method with Lagrange multipliers.* To appear in Numer. Math.

[7] R. TEZAUR, P. VANĚK, AND M. BREZINA, *Two-level method for solids.* In preparation.

[8] P. VANĚK AND J. KŘÍŽKOVÁ, *Two–level method on unstructured meshes with convergence rate independent of the coarse-space size*, Tech. Report 35, Center of Computational Mathematics/UCD, 1995.

[9] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems.* To appear in Computing.

Topic: *Applications*	*Session Chair:* *Tom Russell*	*Room B*
4:45 - 5:15	C. Yang	Numerical Computation of the Linear Stability of the Diffusion Model for Crystal Growth Simulation
5:15 - 5:45	L. Borges	Highly Indefinite Multigrid for Eigenvalue Problems
5:45 - 6:15	G.S. Lett	An Adaptive Nonlinear Solution Scheme for Reservoir Simulation
6:15 - 6:45	A. Cardona	An Iterative Method to Invert the LTSn Matrix

NUMERICAL COMPUTATION OF THE LINEAR STABILITY OF THE DIFFUSION MODEL FOR CRYSTAL GROWTH SIMULATION

C. YANG *, D. C. SORENSEN †, D. I. MEIRON ‡ AND B. WEDEMAN §

Abstract. We consider a computational scheme for determining the linear stability of a diffusion model arising from the simulation of crystal growth. The process of a needle crystal solidifying into some undercooled liquid can be described by the dual diffusion equations

$$\frac{\partial U_l}{\partial t} = \alpha \nabla^2 U_l, \quad \frac{\partial U_s}{\partial t} = \alpha \nabla^2 U_s,$$

with appropriate initial and boundary conditions. Here U_l and U_s denote the temperature of the liquid and solid respectively, and α represents the thermal diffusivity. At the solid-liquid interface, the motion of the interface denoted by $\vec{r}$ and the temperature field are related by the conservation relation

$$\frac{d\vec{r}}{dt} \cdot \vec{n} = \alpha(\nabla U_s \cdot \vec{n} - \nabla U_l \cdot \vec{n}),$$

where $\vec{n}$ is the unit outward pointing normal to the interface. A basic stationary solution to this free boundary problem can be obtained by writing the equations of motion in a moving frame and transforming the problem to parabolic coordinates. This is known as the Ivantsov parabola solution. Linear stability theory applied to this stationary solution gives rise to an eigenvalue problem of the form

$$\begin{aligned} \frac{1}{\eta^2+\xi^2}\left[\frac{\partial^2 U}{\partial \xi^2} + \frac{\partial^2 U}{\partial \eta^2} + 2P\left(\eta\frac{\partial U}{\partial \eta} - \xi\frac{\partial U}{\partial \xi}\right)\right] &= \lambda U, \\ \frac{-1}{1+\xi^2}\left[\frac{\partial U}{\partial \eta} + 4P^2 N + 2P(N + \xi\frac{\partial N}{\partial \xi})\right] &= \lambda N, \\ U = 2PN \quad \text{at } \eta = 1. \end{aligned}$$

The largest real part of the eigenvalue λ is proportional to the growth rate of the perturbation, and the eigenfunction is related to the perturbation of the temperature field and the interface geometry. Numerical solution of the above equations is based on a finite difference discretization. The corresponding large scale algebraic eigenvalue problem is solved by ARPACK, a software package that implements the Implicitly Restarted Arnoldi Method (IRAM.)

Accurate computation of these eigenvalues helps to determine interesting unstable modes that involve excitation of the interface. Analysis suggests that at least part of the spectrum corresponding to this eigenvalue problem is continuous and unbounded. In addition computation via standard methods such as QR becomes expensive when the mesh size of the discretization becomes small. We find however that IRAM is very efficient in extracting eigenvalues and eigenvectors of interest with modest cost. Numerical results will be presented to demonstrate the effectiveness this method.

1. Introduction.

There has been a great deal of interest in the simulation and modeling of crystal growth and dentridic solidification in the past few years [2] [6]. It is well known that the physical behavior of a needle crystal solidifying into some undercooled liquid can be described by the dual diffusion equations

$$\frac{\partial U_l}{\partial t} = \alpha \nabla^2 U_l, \quad \frac{\partial U_s}{\partial t} = \alpha \nabla^2 U_s, \tag{1}$$

* Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA; E-Mail: chao@caam.rice.edu .

† Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA; E-Mail: sorensen@caam.rice.edu.

‡ Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125, USA; E-Mail: dim@ama.caltech.edu.

§ Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125, USA; E-Mail: bw@ama.caltech.edu.

Here U_l and U_s denote the temperature of the liquid and solid respectively. They are functions of the time t and the spatial coordinates x and z. The parameter α represents the thermal diffusivity. At the solid-liquid interface, $U_l = U_s$, and the motion of the interface denoted by $\vec{r}$ and the temperature field are related by the conservation relation

$$\frac{d\vec{r}}{dt} \cdot \vec{n} = \alpha(\nabla U_s \cdot \vec{n} - \nabla U_l \cdot \vec{n}), \tag{2}$$

where $\vec{n}$ is the unit outward pointing normal to the interface. It is also natural to impose the boundary condition

$$U_l \to 0, \text{ as } z \to \infty. \tag{3}$$

Both analytical and numerical solutions of (1) and (2) are difficult to obtain because of the moving boundary. We are interested in analyzing the stability of a well known stationary solution that corresponds to a simple parabolic shaped moving front. In the following, we give a brief description of the Ivantsov solution and a standard linear stability analysis that gives rise to an eigenvalue problem. Numerical discretization of the continuous model and the solution of the large scale algebraic eigenvalue problem derived from the discretization are also discussed. It is observed from our numerical computation that the solidification is unstable.

2. Ivantsov solution. A stationary solution of (1) that corresponds to a parabolic shaped moving front can be obtained by the method of Ivantsov [3]. Suppose the front is moving in the z direction with a constant velocity v. We first rewrite the equation (1) in a moving frame. *i.e.*, we let

$$z \leftarrow z - vt \text{ and } x \leftarrow x. \tag{4}$$

After these changes of variables, equations (1) become

$$\nabla^2 U + \frac{2}{l}\frac{\partial U}{\partial z} = \frac{1}{\alpha}\frac{\partial U'}{\partial t}, \tag{5}$$

in both the liquid and solid phases. The boundary conditions (2) (3) remain the same. To simplify the geometry, transformations

$$x = \rho\eta\xi \text{ and } z = \rho\frac{\eta^2 - \xi^2}{2} \tag{6}$$

are used to map the parabolic interface in (x, z) coordinates to the horizontal line $\eta = 1$ in (ξ, η) coordinates. In these new coordinates, the convection diffusion equation (5) can be written as

$$\frac{\partial^2 U}{\partial \xi^2} + \frac{\partial^2 U}{\partial \eta^2} + 2P\left(\eta\frac{\partial U}{\partial \eta} - \xi\frac{\partial U}{\partial \xi}\right) = (\eta^2 + \xi^2)P\frac{\partial U}{\partial \tau}, \tag{7}$$

where $P \equiv \rho/l$ is the *Peclet* number, and τ is defined to be $\tau \equiv (v/2\rho)t$. The boundary condition imposed at the moving front $\eta = 1$ satisfies

$$P\left[\frac{\partial N}{\partial t}(N^2 + \xi^2) + 2\left(N + \xi\frac{\partial N}{\partial \xi}\right)\right] = \left(\frac{\partial U_s}{\partial \eta} - \frac{\partial U_l}{\partial \eta}\right) - \frac{\partial N}{\partial \xi}\left(\frac{\partial U_s}{\partial \xi} - \frac{\partial U_l}{\partial \xi}\right). \tag{8}$$

It is easy to verify that a stationary solution to (7) and (8) is in the form

$$\bar{N} = 1, \quad \bar{U}_l = \sqrt{\pi P}\exp(P)\operatorname{erfc}(\sqrt{P}\eta), \text{ and } \bar{U}_s = \sqrt{\pi P}\exp(P)\operatorname{erfc}(\sqrt{P}). \tag{9}$$

3. Linear Stability Analysis. The objective of this paper is to determine the linear stability of the Ivantsov solution under small disturbance. This is done by assuming that there exists a solution to (7) and (8) of the form

$$(10)\quad N = \bar{N} + \tilde{N}\exp(\sigma\tau),\quad U_l = \bar{U}_l + \tilde{U}_l\exp(\sigma\tau),\ \text{and}\ U_s = \bar{U}_s + \tilde{U}_s\exp(\sigma\tau),$$

where $\bar{N}$, $\bar{U}_l$ and $\bar{U}_s$ are stationary solutions derived by Ivantsov method, and σ is the growth rate.

The substitution of (10) into (7) leads to the disturbance equation

$$(11)\qquad \left(\frac{\partial^2\tilde{U}}{\partial\xi^2} + \frac{\partial^2\tilde{U}}{\partial\eta^2}\right) + 2P\left(\eta\frac{\partial\tilde{U}}{\partial\eta} - \xi\frac{\partial\tilde{U}}{\partial\xi}\right) = (\eta^2+\xi^2)P\sigma\tilde{U},$$

in both phases with boundary conditions

$$\bar{U}_s = 0 \text{ everywhere},\quad \bar{U}_l = -\frac{\partial\tilde{U}_l}{\partial\eta}\tilde{N} \text{ at } \eta = 1 \text{ and}$$

$$(12)\qquad P[\sigma\tilde{N}(1+\xi^2) + 2(\tilde{N} + \xi\frac{\partial\tilde{N}}{\xi})] = \left(-\frac{\partial\tilde{U}_l}{\partial\eta} - 4P^2\tilde{N}\right),\ \text{at } \eta = 1.$$

To simplify the notation, we rename variables $\tilde{N}$ and $\tilde{U}$ to N and U respectively, and let $\lambda = \sigma P$. Equation (11) and the boundary condition (12) can be written as the following eigenvalue problem:

$$(13)\qquad \frac{1}{\eta^2+\xi^2}\left[\frac{\partial^2 U}{\partial\xi^2} + \frac{\partial^2 U}{\partial\eta^2} + 2P\left(\eta\frac{\partial U}{\partial\eta} - \xi\frac{\partial U}{\partial\xi}\right)\right] = \lambda U$$

$$(14)\qquad -\frac{1}{1+\xi^2}\left[\frac{\partial U}{\partial\eta} + 4P^2N + 2P\left(N + \xi\frac{\partial N}{\partial\xi}\right)\right] = \lambda N\quad(\eta = 1).$$

where U and N are coupled by $U = 2PN$ at $\eta = 1$.

On an infinite domain. The boundary condition at infinity are

$$\frac{\partial U}{\partial\eta} \to 0,\ \text{as}\ \eta \to \pm\infty,\ \text{and}\ \frac{\partial U}{\partial\xi} \to 0,\ \text{as}\ \xi \to \pm\infty.$$

4. Discretization. In our numerical approximation, the infinite domain problem is first transformed into a finite domain problem by using the following change of variables. Let

$$\tilde{s} = \frac{\xi}{1+\xi}\ \text{ and }\ \tilde{t} = \frac{2\eta}{1+\eta}.$$

In these new variables, (13) and (14) become

$$(15)\quad C(\tilde{s},\tilde{t})\left[(1-\tilde{s})^4\frac{\partial^2 U}{\partial s^2} + \frac{1}{4}(2-\tilde{t})^4\frac{\partial^2 U}{\partial t^2} - E(\tilde{s})\frac{\partial U}{\partial\tilde{s}} + F(\tilde{t})\frac{\partial U}{\partial\tilde{t}}\right] = \lambda U,$$

$$(16)\qquad D(\tilde{s},\tilde{t})\left\{P\frac{\partial U}{\partial\tilde{t}} + 4P^2N + 2P\left[N + (1-\tilde{s})^2\frac{\partial N}{\partial\tilde{s}}\right]\right\} = -\lambda N,$$

where

$$C(\tilde{s},\tilde{t}) = \left[(\frac{\tilde{s}}{1-\tilde{s}})^2 + (\frac{\tilde{t}}{2-\tilde{t}})^2\right]^{-1},\quad D(\tilde{s},\tilde{t}) = \left[1 + (\frac{\tilde{s}}{1-\tilde{s}})^2\right]^{-1},$$

$$E(\tilde{s}) = 2(1-\tilde{s})^3 + 2P\tilde{s}(1-\tilde{s}),\ \text{and}\ F(\tilde{t}) = -\frac{1}{2}(2-\tilde{t})^3 + P\tilde{t}(2-\tilde{t}).$$

Let $\tilde{s}_i = i\Delta\tilde{s}$, $\tilde{t}_j = j\Delta\tilde{t}$, $U_{ij} = U(\tilde{s}_i, \tilde{t}_j)$, and $N_i = N(\tilde{s}_i)$. The standard centered difference formula is used to discretize the equation (15). At the boundary $\tilde{s} = 0$ and $\tilde{s} = 1$, we use ghost values $U_{-1,j} = U_{1,j}$, $U_{n+1,j} = U_{n-1,j}$ and centered difference to discretize $\partial U/\partial\tilde{s}$. A similar scheme is used to discretize $\partial U/\partial\tilde{t}$ at $\tilde{t} = 2$. At the interface boundary $\tilde{t} = 1$, the temperature $U_{i,0}$ and the displacement of the moving front N_i satisfies $U_{i,0} = 2PN_i$. To avoid mixing U and N values, an upwind difference scheme is used to discretize the term $\partial U/\partial\tilde{t}$ in (16). The term $\partial N/\partial\tilde{s}$ is approximated by centered difference.

The above discretization scheme gives rise to an algebraic eigenvalue problem

$$Ax = \lambda x, \text{ where}$$

$$x = \begin{bmatrix} U_1 \\ \vdots \\ U_m \\ N \end{bmatrix}, \; U_j = \begin{bmatrix} U_{0,j} \\ \vdots \\ U_{n,j} \end{bmatrix} \text{ and } N = \begin{bmatrix} N_0 \\ \vdots \\ N_n \end{bmatrix}$$

Eigenvalues of positive real parts are sought to determine interesting unstable modes that involve excitation of the interface. Analysis [5] suggests that at least part of the spectrum corresponding to this eigenvalue problem is continuous and unbounded. The conventional QR method become expensive as the mesh size of the discretization gets small. Fast iterative scheme such as the Arnoldi method is attractive in this setting.

5. Implicitly Restarted Arnoldi Method. The standard Arnoldi method computes a factorization of the form

$$AV_k = V_k H_k + f e_k^T, \;\; V_k^H V_k = I, \;\text{ and }\; V_k^H f = 0,$$

where H_k is a $k \times k$ upper Hessenberg matrix. The first column of V_k is arbitrarily chosen and normalized such that $\|v_1\| = 1$. Subsequent columns of V_k, the matrix H_k and the vector f are generated from the Arnoldi process illustrated below.

Input: (A, v_1)
Output: (V_k, H_k, f)
$w \leftarrow Av$; $\alpha_1 = v_1^H w$;
$H_1 = (\alpha_1)$; $V_1 = (v_1)$; $f \leftarrow w - v_1\alpha_1$;
for $j = 1, 2, 3, ..., k-1$

1. $\beta_j = \|f\|$; $v_{j+1} \leftarrow f/\beta_j$
2. $V_{j+1} = (V_j, v_{j+1})$; $\begin{pmatrix} H_j \\ \beta_j e_j^T \end{pmatrix}$;
3. $z \leftarrow Av_{j+1}$;
4. $h \leftarrow V_j^H z$; $H_{j+1} = (H_{j-1}, h)$;
5. $f \leftarrow z - V_{j+1}h$;

end;

It can be verified that the columns of V form an orthonormal basis for the Krylov subspace $\mathcal{K} = \{v_1, Av_1, ..., A^{k-1}v_1\}$. Eigenvalues of H provide approximations to the eigenvalues of A. They are often referred to as the Ritz values. If y is the eigenvector of H corresponding to an eigenvalue θ, the Ritz vector $z = Vy$ is an approximation to an eigenvector of A. It is well known that Ritz values converges very fast to well separated extreme eigenvalues. However, in our problem these eigenvalues correspond

to the ones on the left half of the real axis, and are not interesting. To overcome this difficulty, one must construct a starting vector v_1 such that the subspace spanned by columns of V contains the desired eigencomponents. The construction of v_1 is not trivial. The Implicitly Restarted Arnoldi Method (IRAM) [7] provides an efficient scheme to repeatedly modify an arbitrary starting vector v_1 so that the unwanted eigencomponents v_1 are annihilated by a polynomial in A. The analysis and some of the implementation issues of IRAM are also contained in [4]. The basic theory is outlined below.

Given a $(k+p)$-step Arnoldi factorization

$$AV_{k+p} = V_{k+p}H_{k+p} + fe_{k+p}^T, \quad V_{k+p}^H V_{k+p} = I, \quad V_{k+p}^H f = 0, \tag{17}$$

a sequence of QR updates corresponding to the shifts $\mu_1, \mu_2, ..., \mu_p$ may be applied as follows. Let $H_{k+p} - \mu_1 I = Q_1 R_1$ be the QR decomposition of $H_{k+p} - \mu_1 I$, it follows from (17) that

$$(A - \mu_1 I)V_{k+p} = V_{k+p}(H_{k+p} - \mu_1 I) + fe_{k+p}^T = (V_{k+p}Q_1)R_1 + fe_{k+p}^T. \tag{18}$$

Multiplying the above equation on the right by Q_1 yields

$$(A - \mu_1 I)(V_{k+p}Q_1) = (V_{k+p}Q_1)(Q_1^H H_{k+p}Q_1) + fe_{k+p}^T Q_1.$$

It is easily seen from (18) that the first column of the updated $V_{k+p}^+ = V_{k+p}Q_1$ is related to the first column of V_{k+p} through $(A - \mu_1 I)v_1 = v_1^+ \rho_{11}$. Let $H_{k+p}^+ = Q_{j-1}^H H Q_{j-1}$. The next step starts with the factorization of $H_{k+p}^+ - \mu_2 I$ followed by the update of V_{k+p}^+ and H_{k+p}^+. After all p shifts have been used, the Arnoldi factorization can be recovered by dropping the last p columns of V_{k+p}^+ and H_{k+p}^+ and performing p more steps of Arnoldi iteration to give

$$AV_{k+p}^+ = V_{k+p}^+ H_{k+p}^+ + f^+ e_{k+p}^T.$$

This is equivalent to a new Arnoldi factorization with v_1 replaced by $v_1^+ = P_p(A)v_1$, where $P_p(\lambda)$ is a polynomial with roots at $\mu_1, \mu_2, ..., \mu_p$. This polynomial is designed to filter out the unwanted eigen-components in the original starting vector v_1. Thus the shifts $\mu_1, \mu_2, ..., \mu_p$ are chosen to be approximations to the unwanted eigenvalues of A.

A software package based on this algorithm, ARPACK is used successfully in our computation. Table 1 lists the leading eigenvalues that corresponds to different levels discretization and the number of matrix vector multiplications (MATVECs) and CPU time used to obtain them. The *Peclet* number is set to be 0.1 in our computation. The experiment is performed on a SUN-SPARC 10. For coarse discretization up to about $\Delta\tilde{s} = \Delta\tilde{t} = 1/29$, the results compared favorably to those obtained from the LAPACK [1]. As the matrix size increases, the computation becomes more expensive as indicated by a large number of matrix vector multiplications used. In the case $\Delta\tilde{s} = \Delta\tilde{t} = 1/99$, IRAM did not converge in 300 iterations.

An alternative to compute the eigenvalues of A directly is to work with $(A-\sigma I)^{-1}$, where σ is an estimated location of desired eigenvalue. Since eigenvalues of $(A-\sigma I)^{-1}$ are often large and well separated, the Arnoldi approximation converges extremely fast. However, the fast convergence is obtained at the cost of factoring the matrix $(A-\sigma I)$

matrix size	eigenvalue	MATVECs	CPU(seconds)
2500	6.39	4381	876.68
3600	7.78	6645	1252.61
4900	9.17	10406	2664.33
6400	10.6	10508	3847.50

TABLE 1

The performance of ARPACK in direct mode. Three eigenvalues are found in each run. Parameters k and p are set to be 4 and 40 repectively.

matrix size	LSs	CPU(seconds)
2500	121	44.09
3600	121	69.19
4900	87	73.88
6400	88	107.25
8100	86	147.88
10000	83	188.95

TABLE 2

The performance of ARPACK in shift-invert mode. The shift used is $\sigma = 15.0$. Ten eigenvalues are found in each run. Parameters k and p are set to be 10 and 50 respectively.

and solving a linear system $(A - \sigma I)w = v$ at each iteration. In our application, the matrix can be easily factored using a block Guass elimination. The initial shift can be predicted from the runs of smaller size problems. In Table 2 we list the number of linear system solved (LSs) and the CPU time used for problems of various size. It is observed that using ARPACK in shift-invert mode is considerably faster in this application.

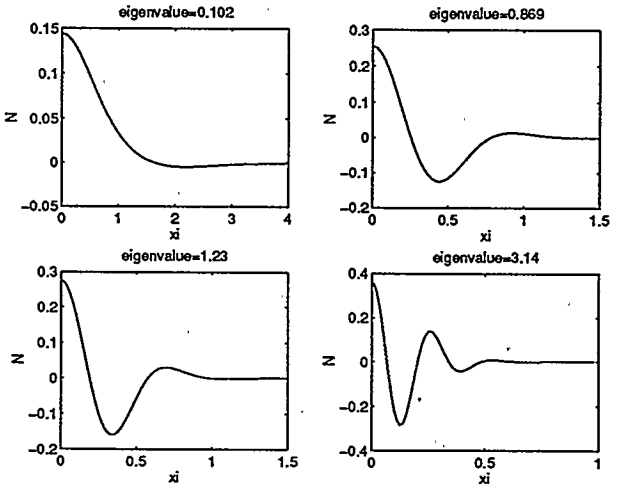


FIG. 1. *The interface N associated with different eigenvalues*

6. Numerical Results. Our computation shows that there are many eigenvalues of A with positive real parts. This implies that the solidification of the needle crystal is unstable. It is also observed in our computation that the leading eigenvalue

increases as $\Delta\tilde{s}$ and $\Delta\tilde{t}$ decrease. This agrees with the analytic prediction that eigenvalues are unbounded as $\Delta\tilde{s}, \Delta\tilde{t} \to 0$. The computed interface N for the disturbance equation corresponding to the four positive eigenvalues of A are plotted in Figure 1. The computation is done on a grid with $\Delta\tilde{s} = \Delta\tilde{t} = 1/99$. It is observed that as the eigenvalue increases, the interface becomes more oscillatory. This agrees with the result obtained from analysis [5]. Finally the temperature field U in both phases that corresponds to a typical positive eigenvalue is plotted in Figure 2.

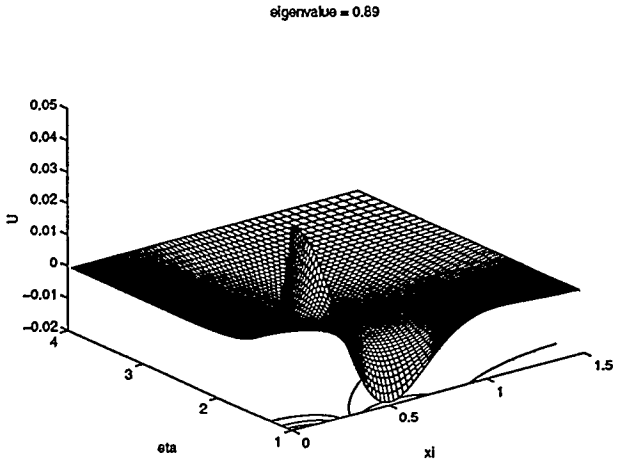


FIG. 2. *The temperature field associated with* $\lambda = 0.869$.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, PA., second edition, 1992.

[2] K. Brattkus and D. I. Meiron. Numerical simulations of unsteady crystal growth. *SIAM Journal on Applied Mathematics*, 52(5):1303–1320, October 1992.

[3] G. P. Ivantsov. Mathematical physics. *Dokl. Akad. Nauk. SSSR*, 58:567, 1947. Translaftted by G. Horvay. Report No. 60-RL-(2511M), G.E. research Lab., Schenectedy, New York, 1960.

[4] R. B. Lehoucq. *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1995.

[5] D. I. Meiron. Private communication, 1996.

[6] J. A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98:231–253, 1992.

[7] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, January 1992.

Highly Indefinite Multigrid for Eigenvalue Problems

by L. Borges and S. Oliveira

1 Introduction

Eigenvalue problems are extremely important in understanding dynamic processes such as vibrations and control systems. Large scale eigenvalue problems can be very difficult to solve, especially if a large number of eigenvalues and the corresponding eigenvectors need to be computed. For solving this problem a multigrid preconditioned algorithm is presented in "The Davidson Algorithm, preconditioning and misconvergence" [2]. Another approach for solving eigenvalue problems is by developing efficient solutions for highly indefinite problems. In this paper we concentrate on the use of new highly indefinite multigrid algorithms (such as in Shapira's 1995 paper [3]) for the eigenvalue problem.

A Newton iteration coupled with the algorithm of Shapira [3] for solving the highly indefinite linear systems can generate efficient iterative methods for solving eigenvalue problems. Notice that the original guess plays a special role on the efficiency of the algorithm. In fact, it has been observed already that if the correct spectrum is well spaced out, this algorithm works better than standard algorithms such as Rayleigh quotient methods. We are in the process of analyzing the use of diverse initial guesses for this iterative eigenvalue algorithm.

This Newton-multigrid algorithm has the advantage of being highly parallelizable. Consider an environment in which an unlimited number of processors is available; each processor could address the search of one of the eigenvalues of the matrix which makes this approach attractive. Coordination between processors is only needed to prevent solutions on different processors from converging to the same eigenvalue and eigenvector. Methods for achieving this are under development.

It is relevant to mention that the multigrid for highly indefinite systems of Shapira does not address all the cases of interest where eigenvalue problems arise. Thus we are also working on modifications of these algorithms. At this stage our computational results are promising but our theoretical results are preliminary. Most of this work will be shown in the conference presentation.

2 A Newton–Multigrid Algorithm for Eigenvalues

Let A be a square matrix and consider (λ, x) as the approximate eigenvalue and eigenvector pair. The associated pair $(\delta\lambda, \delta x)$ which corrects λ and x, respectively, must satisfy

$$[A - (\lambda + \delta\lambda)I](x + \delta x) = 0 \, .$$

Omitting the small term $\delta\lambda\ \delta x$ this becomes

$$(A - \lambda I)\delta x = \delta\lambda x - (A - \lambda I)x\ . \tag{1}$$

Adopting the approximated normalized eigenvectors: $x^T x = 1$ and $(x+\delta x)^T(x+\delta x) = 1$ in (1) and considering $||\delta x||$ as an small term, we have

$$x^T \delta x + \delta x^T x = 0 \Rightarrow\ x^T \delta x = 0. \tag{2}$$

Let v the solution for

$$(A - \lambda I)v = x\ . \tag{3}$$

From equation (1), we have

$$\delta x = \delta\lambda\,(A - \lambda I)^{-1}x - x = \delta\lambda\, v - x.$$

Thus, the final eigenvector is given by

$$x + \delta x = \delta\lambda v\ , \tag{4}$$

that is, $(x + \delta x)$ must be parallel to v. Also

$$x^T \delta x = x^T v\,\delta\lambda - x^T x\ .$$

And using (2),

$$\delta\lambda = \frac{x^T x}{x^T v} = \frac{1}{x^T v}. \tag{5}$$

Equations (3), (4) and (5) lead to the iterative algorithm

Eigenvalue Algorithm

Given a normalized initial guess x.
$\lambda = x^T A x$ /* Rayleigh quotient */
while $||Ax - \lambda x|| \geq \epsilon$
$v = (A - \lambda I)^{-1}x$
$\delta\lambda\ = 1/x^T v$
$\lambda = \lambda + \delta\lambda$
$x\ = v/||v||$
endwhile

Notice that this algorithm is different from the Rayleigh quotient method [1] since it iteratively corrects the eigenvalue λ using $\delta\lambda$ instead of the Rayleigh quotient. The behavior of this algorithm may cause misconvergence, but on the other hand, when the corrections δx and $\delta\lambda$ are small enough, each new correction $\delta\lambda$ captures the actual error between the real eigenvalue and its approximation λ, and the method converges fast as is the case with Newton derived methods. Numerical results are presented in Section 4.

3 Multigrid Methods and Eigenvalue Problems

Since part of this algorithm uses an approach close to inverse iteration, a critical point is the solution of either highly indefinite or nearly singular systems. When considering a multigrid solver for (3), we must attempt to guarantee a minimal set of properties as follows.

Since the linear system solver is supposed to be generic as much as possible, the multigrid method adopted must serve as an "automatic" solver depending only on the finest grid equations. That means, coarse grid, restriction and prolongation operators are easily obtained from the original linear system. Moreover, the solver must deal with highly indefinite equations.

We have adopted the AutoMUG method [3, 4] as the multigrid solver because it is designed to achieve many of these properties. Perhaps the critical difference lies in its restriction to symmetric pentadiagonal matrices. Even so, it seems to be very attractive as a first choice.

Our first step is to bring the AutoMUG concept back to directly using a Schur complement approach. See [3] for a complete description of the method.

Red–black (RB) ordering provides a natural way to parallelize linear systems arising from $(2d+1)$ point stencils for discretization of partial differential equations in d dimensions. Thus the system can be split into two problems: red-system and black-system by adopting a Schur complement method. For the one dimensional case, let

$$Ax = b \tag{6}$$

where A is a tridiagonal matrix of order N. Suppose A has no vanishing diagonal element and let $D = \mathrm{diag}(A)$. We can rewrite

$$A = DM^t \begin{pmatrix} I & -B \\ -C & I \end{pmatrix} M \ ,$$

where $M = M(K)$ is the permutation matrix which reorders the variables of a K-dimensional vector such that odd variables appear in the first block and even numbered variables appear in the second block. Thus, B and C are the bidiagonal matrices:

$$C = -\mathrm{tridiag}(0, \frac{a_{2i,2i-1}}{a_{2i,2i}}, \frac{a_{2i,2i+1}}{a_{2i,2i}})_{1 \le i \le \lfloor K/2 \rfloor}$$

and

$$B = -\mathrm{tridiag}(\frac{a_{2i-1,2i-2}}{a_{2i-1,2i-1}}, \frac{a_{2i-1,2i}}{a_{2i-1,2i-1}}, 0)_{1 \le i \le \lceil K/2 \rceil} \cdot$$

For a diagonal matrix D let

$$even(D) = \mathrm{diag}(d_{2i})_{1 \le i \le \lfloor K/2 \rfloor} \quad \text{and} \quad odd(D) = \mathrm{diag}(d_{2i-1})_{1 \le i \le \lceil K/2 \rceil} \cdot$$

So, rewriting (6):

$$\begin{pmatrix} odd(D) & -odd(D)B \\ -even(D)C & even(D) \end{pmatrix} \begin{pmatrix} x_{odd} \\ x_{even} \end{pmatrix} = \begin{pmatrix} b_{odd} \\ b_{even} \end{pmatrix} \tag{7}$$

Consider the two-level scheme for residual correction in (6):

$$\begin{array}{ll} \text{a.} & \textit{guess}\ \ x_{in} \\ \text{b.} & \textit{solve}\ \ Ae = r \doteq Ax_{in} - b \\ \text{c.} & x_{out} = x_{in} - e \end{array} \tag{8}$$

One possible approach is obtained using a Schur Complement method over the system (8b) where the block decomposition is induced by the odd-even ordering. This procedure that results consists of the following steps:

$$\begin{array}{ll} \text{a.} & odd(D)\bar{e}_{odd} = r_{odd} \\ \text{b.} & Qe_{even} = Rr \quad \text{where} \quad Q = even(D)(I - CB), \\ & \qquad\qquad\qquad\quad R = (\tilde{C}\ \ I)M, \\ & \qquad\qquad\qquad\quad \tilde{C} = even(D)Codd(D)^{-1} \\ \text{c.} & e_{odd} = \bar{e}_{odd} + Be_{even} \end{array} \tag{9}$$

Notice that the system (9b) has the same order of the even block.
Omitting (9a), that is, $\bar{e}_{odd} = 0$, (8c) is written like

$$x_{out} = x_{in} - M^t \begin{pmatrix} e_{odd} \\ e_{even} \end{pmatrix} = x_{in} - M^t \begin{pmatrix} Be_{even} \\ e_{even} \end{pmatrix} = x_{in} - Pe_{even}$$

where $P = M^t \begin{pmatrix} B \\ I \end{pmatrix}$.

Thus (8) leads to a two level algorithm

$$\begin{array}{ll} \text{a.} & \textit{guess}\ \ x_{in} \\ \text{b.} & e = 0 \\ \text{c.} & \textit{solve}\ \ Qe = R(Ax_{in} - b) \\ \text{d.} & x_{out} = x_{in} - Pe \end{array} \tag{10}$$

which can be seen as a multigrid algorithm with coarse grid, restriction and prolongation operators (Q, R and P respectively) such that $Q = RAP$. Since the product CB is tridiagonal, so is Q. It makes possible to define a multilevel solver based on (10) and (9b).

To apply this algorithm for two-dimensional problems we need, for example, to reduce a five points stencil to a three points stencil notation, as in the one-dimensional case. It can be achieved using an permutation matrix U such that for any vector v defined on a grid and ordered lexicographically row by row, Uv is the same vector v ordered lexicographically column by column. Thus A is written as

$$A = X + U^t Y U\ ,$$

and the previous algorithm applies to the X and Y components of A.

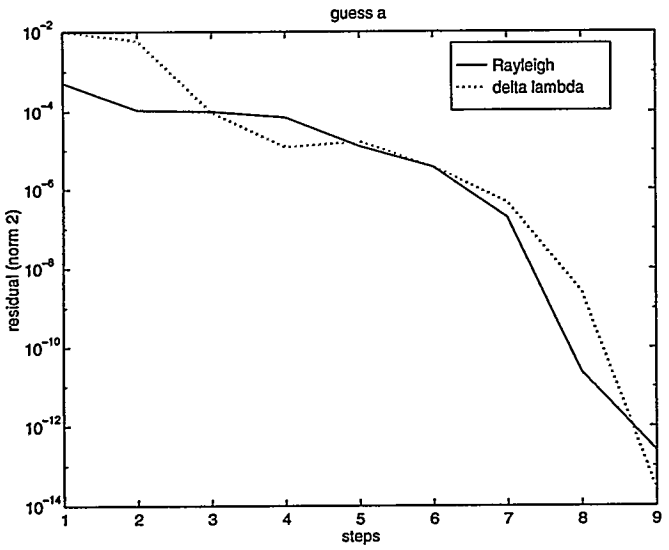


Figure 1: Results for guess a

4 Numerical experiments

We have performed numerical experiments using Matlab^{TM}. It provides a faster way to implement prototypes and compare the results with theoretical estimates. Thus, in the first two figures, solely to test the Newton iteration algorithm, we resort to Matlab routines.

Our algorithm was presented in Section 2. In each of these figures we show the residual $Ax - \lambda x$ using the same initial guess, applied to both algorithms, namely the Rayleigh quotient method and the Newton iteration (delta lambda).

Results for guess a are shown in Figure 1 and guess b in Figure 2. A is a 225×225 pentadiagonal matrix obtained for a second-order central difference scheme for the operator $u_{xx} + u_{yy} + \beta u$ in the unit square. (Here $\beta = 200.$) A direct solver is used in equation (3) for both methods. In the same way Figure 2 corresponds to another guess b.

Both algorithms show the same behavior in the convergence curves but the new version may be delayed by some steps, as illustrated in Figure 2. This effect results from the first group of $\delta\lambda$ estimates: table 1 compares the $\delta\lambda$ estimative with the difference between the nearest eigenvalue $\tilde{\lambda}$ of A and the approximation λ evaluated. In this example, steps 1 and 2 produce $\delta\lambda$ corrections that may move the iterative λ nearer to an different eigenvalue of A, resulting in a two step delay in convergence. Indeed, in case (a) both algorithms converge to the same eigenvalue but it is not true for case (b).

Following the same iterative approach, but using a BiCGSTAB multigrid

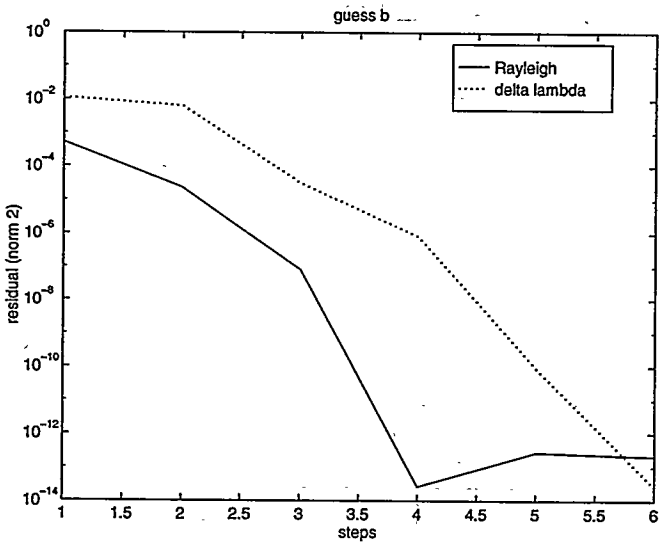


Figure 2: Results for guess b

step	guess a		guess b	
	$\tilde{\lambda} - \lambda$	$\delta\lambda$	$\tilde{\lambda} - \lambda$	$\delta\lambda$
1	7.7e-05	-9.8e-03	6.1e-05	1.1e-02
2	9.3e-05	9.7e-03	-7.1e-05	-1.1e-02
3	-1.5e-05	-1.0e-04	2.2e-05	4.4e-05
4	7.3e-05	8.1e-05	-2.2e-05	-2.2e-05
5	-7.6e-06	-1.8e-05	3.2e-08	3.2e-08
6	1.1e-05	1.2e-05	-2.8e-13	-2.4e-13
7	-1.8e-06	-1.9e-06		
8	9.9e-08	9.9e-08		
9	-6.8e-11	-6.8e-11		

Table 1: Eigenvalue errors and corrections

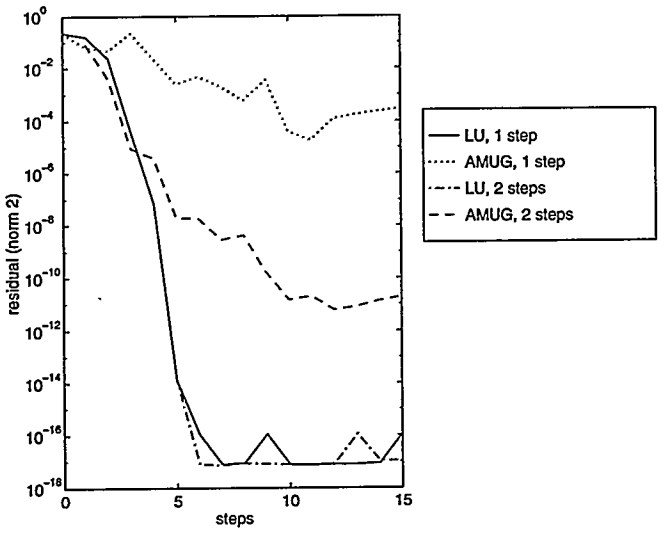


Figure 3: Newton-Multigrid algorithm behavior

preconditioned for the solution of the linear systems (3), we obtain the results shown in Figure 3. This Figure also compares this against the case where the preconditioner is the LU factorization.

The matrix for this example was chosen because of the rather dense concentration of the spectrum. When this is not the case, the Newton-Multigrid algorithm behaves much better. In fact, the graphs for the two kind of solvers (multigrid/LU - BiCGSTAB) basically coincide for many cases.

References

[1] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins Press, 2nd edition, 1989. 1st edition published 1983 by North Oxford Academic.

[2] S. Oliveira. The Davidson algorithm, preconditioning and misconvergence. Submitted for *Algebraic Multilevel Methods and its Applications Conference*, Nijmegen, the Netherlands, June 1996, 1996.

[3] Y. Shapira. Multigrid methods for highly indefinite equations. In S.F. McCormick and T.A. Manteuffel, editors, *Proceedings, 8th Copper Mountain Conference on Multigrid Methods.* NASA, 1995. In press.

[4] Y. Shapira, M. Israeli, and A. Sidi. Towards automatic multigrid algorithms for SPD, nonsymmetric and indefinite problems. *SIAM J. Sci. Computing*, 1996. Accepted for publication.

An Adaptive Nonlinear Solution Scheme for Reservoir Simulation

G. Scott Lett
slett@ssii.com
Scientific Software - Intercomp, Inc.
1801 California Street, Suite 295
Denver, Colorado USA 80202-2699

Numerical reservoir simulation involves solving large, nonlinear systems of PDE with strongly discontinuous coefficients. Because of the large demands on computer memory and CPU, most users must perform simulations on very coarse grids. The average properties of the fluids and rocks must be estimated on these grids. These coarse grid "effective" properties are costly to determine, and risky to use, since their optimal values depend on the fluid flow being simulated. Thus, they must be found by trial-and-error techniques, and the more coarse the grid, the poorer the results.

This paper describes a numerical reservoir simulator which accepts fine scale properties and automatically generates multiple levels of coarse grid rock and fluid properties. The fine grid properties and the coarse grid simulation results are used to estimate discretization errors with multilevel error expansions. These expansions are local, and identify areas requiring local grid refinement. These refinements are added adaptively by the simulator, and the resulting composite grid equations are solved by a nonlinear Fast Adaptive Composite (FAC) Grid method, with a damped Newton algorithm being used on each local grid. The nonsymmetric linear system of equations resulting from Newton's method are in turn solved by a preconditioned Conjugate Gradients-like algorithm.

The scheme is demonstrated by performing fine and coarse grid simulations of several multiphase reservoirs from around the world.

An Iterative Method to Invert the LTSn Matrix

Augusto V. Cardona - PhD Student
E-Mail avcardona@music.pucrs.br

Marco T. de Vilhena
Nuclear Engineering Department - UFRGS
Porto Alegre - Brazil

Recentely Vilhena and Barichello proposed the LTSn method to solve, analytically, the Discrete Ordinates Problem (Sn problem) in transport theory. The main feature of this method consist in the application of the Laplace transformto the set of Sn equations and solve the resulting algebraic system for the transport flux. Barichello solve the linear system containing the parameter s applying the definition of matrix invertion exploiting the structure of the LTSn matrix. In this work, it is proposed a new scheme to invert the LTSn matrix, decomposing it in blocks and recursively inverting this blocks.

Topic: *Helmholtz*	*Session Chair:* *Roland Freund*	*Room C*
4:45 - 5:15	E. Larsson	Iterative Solution of the Holmholtz Equation
5:15 - 5:45	S. Kim	Iterative Procedures for Wave Propagation in the Frequency Domain
5:45 - 6:15	J. Yoo	Multigrid for the Galerkin Least Squares Method in Linear Elasticity: The Pure Displacement Problem
6:15 - 6:45		

Iterative solution of the Helmholtz equation

We have shown that the numerical solution of the two-dimensional Helmholtz equation can be obtained in a very efficient way by using a preconditioned iterative method.
We discretize the equation with second-order accurate finite difference operators and take special care to obtain non-reflecting boundary conditions. We solve the large, sparse system of equations that arises with the preconditioned restarted GMRES iteration. The preconditioner is of "fast Poisson type", and is derived as a direct solver for a modified PDE problem.The arithmetic complexity for the preconditioner is $\mathcal{O}(n \log_2 n)$, where n is the number of grid points.
As a test problem we use the propagation of sound waves in water in a duct with curved bottom. Numerical experiments show that the preconditioned iterative method is very efficient for this type of problem. The convergence rate does not decrease dramatically when the frequency increases. Compared to banded Gaussian elimination, which is a standard solution method for this type of problems, the iterative method shows significant gain in both storage requirement and arithmetic complexity. Furthermore, the relative gain increases when the frequency increases.

Elisabeth Larsson,
Kurt Otto,
Uppsala university, Sweden

Address: Dept. of Scientific Computing
Box 120
S-751 04 Uppsala
SWEDEN
Email address: Elisabeth.Larsson@tdb.uu.se

Iterative Procedures for Wave Propagation in The Frequency Domain

Seongjai Kim* and William W. Symes†

Abstract. A parallelizable two–grid iterative algorithm incorporating a domain decomposition (DD) method is considered for solving the Helmholtz problem. Since a numerical method requires choosing at least 6 to 8 grid points per wavelength, the coarse–grid problem itself is not an easy task for high frequency applications. We solve the coarse–grid problem using a nonoverlapping DD method. To accelerate the convergence of the iteration, an artificial damping technique and relaxation parameters are introduced. Automatic strategies for finding efficient parameters are discussed. Numerical results are presented to show the effectiveness of the method. It is numerically verified that the rate of convergence of the algorithm depends on the wave number *sub–linearly* and does not deteriorate as the mesh size decreases.

1 Introduction

Let $\Omega = (0,1)^r$, $r = 2$ or 3, and $\Gamma = \partial\Omega$. Consider the following Helmholtz problem

$$\begin{aligned} -\Delta u - K(x)^2 u &= f(x), \quad x \in \Omega, \\ \frac{\partial u}{\partial \nu} + i\alpha(x)u &= 0, \qquad x \in \Gamma, \end{aligned} \tag{1}$$

where i is the imaginary unit, ν the unit outward normal from Γ, and the coefficients $K(x)$ and $\alpha(x)$ satisfy

$$K^2 = p^2 - iq^2, \ 0 < p_0 \le p(x) \le p_1 < \infty, \ 0 \le q_0 \le q(x) \le q_1 < \infty,$$
$$\alpha = \alpha_r - i\alpha_i, \quad \alpha_r > 0, \ \alpha_i \ge 0.$$

The coefficient α is properly chosen such that the second equation of (1) represents a first–order absorbing boundary condition that allows normally incident waves to pass out of Ω transparently. Problem (1) models the propagation of time–harmonic waves, discretizations of the time–dependent Schrödinger equation by implicit difference schemes, inverse scattering problems, seismic waves, and ocean acoustics.

Let problem (1) be discretized by a FD/FE method of a mesh size $h = 1/(n-1)$, for an integer $n \ge 1$. We will consider the 2–dimensional case for simplicity ($r = 2$). Then, the approximate solution $\mathbf{u}$ of (1) can be obtained by solving the following algebraic system:

$$A\mathbf{u} = \mathbf{b}, \tag{2}$$

where A is a complex–valued symmetric (but, not Hermitian) $N \times N$ matrix, $N = n^r$, and $\mathbf{b}$ be the source vector. Since we need to choose at least 6 to 8 grid points per wavelength for a stability

*Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251–1892 skim@caam.rice.edu †symes@caam.rice.edu

reason, the dimension of A is often huge for realistic problems. For high–frequency applications, in practice, it is required to choose 12 to 24 grid points per wavelength for accuracy reasons.

For ocean acoustics, we often consider $K(x) = \omega/c(x) = \alpha(x)$, where ω is the angular frequency and c is the wave speed. (In the earth, $c(x)$ has its values between 0.5 Km/sec and 10 Km/sec.) For linear, isotropic, homogeneous electromagnetic waves, we have $K^2 = \mu\varepsilon\,\omega^2 - i\mu\sigma\omega$, where ε is the dielectric permittivity, μ denotes the magnetic permeability, and σ is the electric conductivity. Note that $\mu\varepsilon\,\omega^2 \ll \mu\sigma\omega$ for earth materials at frequencies less than $\texttt{freq} = 10^5$ Hz. (Here $\omega = 2\pi \cdot \texttt{freq}$.) The resulting algebraic system obtained by an approximation technique is therefore diagonally dominant for lower frequency applications and of the form: $A\mathbf{u} = (P + iR)\mathbf{u} = \mathbf{b}$, where P and R are symmetric with R being positive definite; it can be solved relatively easier (in the sense of the rate of convergence) than that of acoustic wave equations. It should be noticed that the existence of a (convergent) nonsymmetric conjugate gradient–type algorithm for (2) is equivalent to the positive definiteness of R [4, 6, 5].

In this article, we consider the *nearly* non–attenuative acoustic waves:

$$p(x) = \frac{\omega}{c(x)} = \alpha(x), \quad 0 \le q \ll p. \tag{3}$$

(In geophysical applications, the *quality factor* Q $(:= p^2/q^2)$ is known to be between 25 and 1000. For the case $q = 0$, we set $Q = \infty$.) In this case, problem (1) is hard to solve numerically. Addition to having a complex–valued solution, it is neither Hermitian–symmetric nor coercive, and the matrix A in (2) is no longer diagonally dominant. As a consequence, most standard iterative algorithms (relaxation methods and conjugate gradient (CG)–type iterative algorithms) either fail to converge or converge so slowly to be impractical. Furthermore, any local source/error invokes an oscillatory solution over the whole domain. (This is another physical reason that relaxation algorithms are hardly convergent.)

Concerning iterative numerical solvers for solving (1), we refer to Bayliss, Goldstein, and Turkel [1] and Freund [5] for the CG–type algorithms. Després [3] studied a domain decomposition (DD) method in a differential level and Kim [7, 8, 9] analyzed nonoverlapping DD methods for solving (1) by finite differences and finite element methods. A heuristic, efficient strategy for choosing the relaxation parameter and an iterative artificial damping technique (IADT) were proposed in [7] and [8], respectively, to obtain a better conditioning of the algorithm. Employing the techniques, the algorithms converge independently on the mesh size h (but, dependently on the number of subdomains) and they can be parallelized efficiently. The authors studied DD methods incorporating the cell–centered finite difference methods [10].

It is known [2] that for sufficiently smooth p the finite element approximation errors are

$$\mathcal{O}(p^{s+2}h^{s+1}),$$

where s is the order of the polynomials used for basis functions. As a result, the number of points per wavelength will have to increase with the wave number to maintain a given accuracy. This makes the problem more difficult, in particular, for high frequency applications. In this paper, we will propose a two–grid algorithm incorporating the DD method in [9] as the coarse grid problem solver.

2 The algorithm

Let (1) be discretized by an approximation scheme, e.g. finite elements or finite difference methods, on a finite dimensional space V^h, where h is the parameter of the mesh size. Then the

resulting algebraic system can be written as

$$A^h \mathbf{u}^h = \mathbf{b}^h, \tag{4}$$

where A is a complex–valued square symmetric (but non–Hermitian) matrix, $\mathbf{u}$ is the unknown vector, and $\mathbf{b}$ denotes the source vector.

Let $H > h$ be the element size for a coarser grid mesh and V^H be the corresponding (proper) subspace of V^h. We consider the following two–grid algorithm:

$$\begin{aligned}
&\text{select } \mathbf{u}^{h,0},\ \varepsilon\,; \\
&\text{do } \ell = 0, 1, \cdots \\
&\quad \text{(i)} \quad \mathbf{r}^{h,\ell} = \mathbf{b}^h - A^h \mathbf{u}^{h,\ell}\,; \\
&\quad \text{(ii)} \quad \text{if } \|\mathbf{r}^{h,\ell}\| > \varepsilon,\ \text{continue}\,; \\
&\quad \text{(iii)} \quad \text{find } \mathbf{e}^H \text{ such that } A^H \mathbf{e}^H = I_h^H(\mathbf{r}^{h,\ell})\,; \\
&\quad \text{(iv)} \quad \mathbf{u}^{h,\ell+1/2} = \mathbf{u}^{h,\ell} + I_H^h(\mathbf{e}^H)\,; \\
&\quad \text{(v)} \quad \mathbf{u}^{h,\ell+1} = S\,(\mathbf{u}^{h,\ell+1/2})\,; \\
&\text{enddo}
\end{aligned} \tag{5}$$

Here $\|\cdot\|$ denotes a norm defined on V^h, e.g. ℓ^2–norm $\|\cdot\|_2$ or ℓ^∞–norm $\|\cdot\|_\infty$. The operators I_h^H and I_H^h denote the projection from V^h to V^H and the interpolation from V^H to V^h, respectively, and S is a smoothing operator. The above two–grid algorithm is the simplest case of MG methods. A MG algorithm can be obtained by trying to solve (5.iii) in another proper subspace $V^{H'} \subset V^H$ corresponding to another coarser grid mesh. A step of MG method consists of two substeps: a coarse grid correction ((5.iii) and (5.iv)) and a smoothing step (5.v). The reason why the MG method is so efficient is, roughly speaking, the following. In the coarse grid correction the low and medium frequency components of the error (corresponding to small and medium eigenvalues) are significantly reduced, while the smoothing step reduces the high frequency components of the error.

There are technical difficulties in applying two–grid methods to problem (1) that is complex–valued and non–coercive. The convergence analysis for two–grid methods is often based on coercivity analyses. Another difficulty in the simulation of wave propagation by two–grid methods, in particular, is in the coarse grid correction: since one needs to choose at least 6 to 8 grid points per wavelength ($2\pi/p$) for stability and/or accuracy reasons, it is hard to solve the problem on the coarse grid for large wave numbers. The efficiency of the two–grid algorithm depends strongly on the solution procedure for the coarse grid problem. For high frequency wave propagation (i.e. p is large), the grid size h can be chosen such that ph is 1/3 to 1/4. This choice of h is the same as choosing 18 to 24 points per wavelength and it is often required for accuracy reasons in wave propagation simulation.

Now, we consider a method of solving (5.iii). It is known that if the (coarse grid) mesh is not fine enough, problem (5.iii) will be unstable. Moreover, the solution, if any, may have few characteristics of the original physical problem. In the case, (5) may converge slowly or fail to converge; we should choose the coarse grid mesh sufficiently fine.

Let $\{\Omega_j, j = 1, 2, \cdots, M\}$ be a partition of Ω:

$$\overline{\Omega} = \cup_{j=1}^M \overline{\Omega}_j; \quad \Omega_j \cap \Omega_k = \emptyset,\ j \neq k; \quad \Gamma_j = \Gamma \cap \partial\Omega_j; \quad \Gamma_{jk} = \Gamma_{kj} = \partial\Omega_j \cap \partial\Omega_k.$$

Decompose problem (1) over the partition:

$$\begin{aligned} &\text{(a)} \quad -\Delta u_j - K^2 u_j = f(x), && x \in \Omega_j, \\ &\text{(b)} \quad \frac{\partial u_j}{\partial \nu_j} + i\alpha u_j = 0, && x \in \Gamma_j, \\ &\text{(c)} \quad \frac{\partial u_j}{\partial \nu_j} + i\beta u_j = -\frac{\partial u_k}{\partial \nu_k} + i\beta u_k, && x \in \Gamma_{jk}, \end{aligned} \tag{6}$$

where β, $\mathrm{Re}(\beta) > 0$, is a relaxation parameter. Equation (6c) is called the Robin interface boundary condition (RIBC) which impose the continuity of the pressure u and the normal component of its flux on the interfaces. Now, we define the DD iteration as follows:
Given $\{u_j^0\}$, $u_j^0 \in V_j := H^1(\Omega_j)$, find $\{u_j^n\}$, $u_j^n \in V_j$, $n \geq 1$, by solving

$$\begin{aligned} &\text{(a)} \quad -\Delta u_j^n - K^2 u_j^n = f(x), && x \in \Omega_j, \\ &\text{(b)} \quad \frac{\partial u_j^n}{\partial \nu_j} + i\alpha u_j^n = 0, && x \in \Gamma_j, \\ &\text{(c)} \quad \frac{\partial u_j^n}{\partial \nu_j} + i\beta u_j^n = -\frac{\partial u_k^{n-1}}{\partial \nu_k} + i\beta u_k^{n-1} && x \in \Gamma_{jk}. \end{aligned} \tag{7}$$

Després [3], indebted to Lions [11], studied the convergence of algorithm (7) in differential, rather than discrete, level. The algorithm was applied to finite differences and finite element approximations by Kim [7, 9] and the cell–centered finite difference method by the authors [10].

Our next objective is to introduce a finite element procedure for algorithm (7). Note that the RIBC (7c) imposes the continuity of both the solution u and the normal components of its flux, while most finite element methods admit discontinuity of the normal components of the flux. Let $\mathcal{T}_h$ be a rectangular (resp., cube–shaped) triangulation of Ω with the mesh size h, and V^h be the finite element space corresponding to the bi–linear (resp., tri–linear) finite element method on $\mathcal{T}_h$. We assume that the subdomains Ω_j share their boundaries with the elements in $\mathcal{T}_h$. Let the subspace V_j^h be the restriction of V^h onto Ω_j. (For simplicity, we have used h to denote the coarse mesh size instead of H.)

Then, a corresponding FE iterative algorithm for (7) can be defined as follows: Given $u_j^{h,0} \in V_j^h$, $j = 1, \ldots, M$, build $u_j^{h,n} \in V_j^h$, $n \geq 1$, recursively by solving

$$\begin{aligned} &(\nabla u_j^{h,n}, \nabla v)_{\Omega_j} - (K^2 u_j^{h,n}, v)_{\Omega_j} + \left\langle i\alpha u_j^{h,n}, v \right\rangle_{\Gamma_j} + \sum_k \langle \frac{1}{2}(-\frac{\partial u_j^{h,n}}{\partial \nu_j} + i\beta u_j^{h,n}), v \rangle_{\Gamma_{jk}} \\ &\quad = \sum_k \langle \frac{1}{2}(-\frac{\partial u_k^{h,n}}{\partial \nu_k} + i\beta u_k^{h,n-1}), v \rangle_{\Gamma_{jk}} + (f, v)_{\Omega_j}, \quad v \in V_j^h. \end{aligned} \tag{8}$$

Let $|\cdot|_{0,\Gamma_{jk}}$ be the L^2 norm on Γ_{jk} and C_1 and C_2 satisfy, $\forall v \in V_j^h$, $j = 1, \cdots, M$,

$$\sum_j \sum_k \left| \frac{\partial v}{\partial \nu_j} \right|^2_{0,\Gamma_{jk}} \leq C_1 h^{-1} \sum_j (\nabla v, \nabla v)_{\Omega_j} \quad \text{and} \quad \sum_k |v|^2_{0,\Gamma_{jk}} \leq C_2 h^{-1} \| v \|^2_{0,\Omega_j}.$$

Choose $\beta = \beta_r - i\beta_i$ as

$$\beta_i = \frac{C_1 h^{-1}}{4}, \quad \beta_r = \xi \beta_i \frac{p_1^2}{q_0^2}, \tag{9}$$

where $\xi > 1$.

Theorem 1. *Let* $\max_{j,k} |\Gamma_{jk}| = \mathcal{O}(h)$ *and* $\xi = 1 + \left(1 + \dfrac{8h^2 q_0^4}{C_1 C_2 p_1^2}\right)^{1/2}$. *Then the spectral radius of the iteration matrix* $\mathcal{A}$ *of* (8) *is minimized and bounded as*

$$\rho(\mathcal{A}) \leq 1 - C_3 \frac{q_0^4}{p_1^2} h^2, \tag{10}$$

for some $C_3 > 0$ *independent of* h, p, *and* q.

In Theorem 1, the convergence of algorithm (8) is strongly dependent on (and most sensitive to) the positivity of q. In fact, for $q = 0$ we do not know the convergence of the algorithm. To overcome the difficulty of (8) with (3), we will consider a technical scheme, named by the *iterative artificial damping technique* (IADT).

Choose $u^{h,0}$ and $\eta > 0$, find $u^{h,m}$, $m \geq 0$, by solving

$$\begin{aligned} (\nabla u^{h,m+1}, \nabla v)_\Omega - (K^2 u^{h,m+1}, v)_\Omega + \langle i\alpha u^{h,m+1}, v\rangle_\Gamma + (i\eta^2 u^{h,m+1}, v)_\Omega \\ = (f, v)_\Omega + (i\eta^2 u^{h,m}, v)_\Omega, \quad v \in V^h. \end{aligned} \tag{11}$$

Theorem 2. *Let* A^h *be non–singular. Then,* (11) *converges for all* $\eta > 0$.

Remark. In the algorithm (11), the problem is how one can solve steps efficiently. Since each step is better conditioned, one can find more easily a convergent iterative algorithm. We do not have to solve it directly/completely, in practical simulations. The problem can be solved *incompletely* using an iterative method such as (8). (The DD method is used as the inner loop solver.) We will force the inner loop to stop in a certain number of iterations n_*. The following is proposed in [8]:

For given $u_j^{h,0,0} \in V_j^h$, $j = 1, \cdots, M$, choose parameters $\eta > 0$ and $n_* \geq 1$ and build $u_j^{h,m,n} \in V_j^h$, $m \geq 0$, by solving

$$\begin{aligned} \text{(a)}\quad & (\nabla u_j^{h,m,n}, \nabla v)_{\Omega_j} - ((K^2 - i\eta^2)\, u_j^{h,m,n}, v)_{\Omega_j} + \left\langle i\alpha u_j^{h,m,n}, v\right\rangle_{\Gamma_j} \\ & + \sum_k \langle \frac{1}{2}(-\frac{\partial u_j^{h,m,n}}{\partial \nu_j} + i\beta u_j^{h,m,n}), v\rangle_{\Gamma_{jk}} \\ & = \sum_k \langle \frac{1}{2}(-\frac{\partial u_k^{h,m,n}}{\partial \nu_k} + i\beta u_k^{h,m,n-1}), v\rangle_{\Gamma_{jk}} \\ & + (i\eta^2\, u_j^{h,m,0}, v)_{\Omega_j} + (f, v)_{\Omega_j}, \quad v \in V_j^h, \quad n = 1, \cdots, n_*, \\ \text{(b)}\quad & u_j^{h,m+1,0} = u_j^{h,m,n_*}. \end{aligned} \tag{12}$$

Let the domain in 2–D be decomposed into M_x and M_y subdomains in x and y directions, respectively. It is numerically verified that the choices

$$n_* = \frac{M_x + M_y}{4} \sim \frac{M_x + M_y}{5}, \quad \eta = \frac{p}{3} \sim \frac{p}{5} \tag{13}$$

give a (fast) *convergence*. For attenuative waves, i.e., when $q > 0$, algorithm (12) with (13) converges 2–10 times faster than (8). Note that in (10), the most sensitive component to the convergence rate of the DD method is the imaginary part of the square of the wave number. (Here it is $q^2 + \eta^2$ and positive.)

$1/h = 1080,\ H = 3h,\ M_x = 60$					$\texttt{freq} = 20,\ H = 2h,\ M_x = 50$				
freq	ℓ	N_{DD}	r_∞^ℓ	CPU	$1/h$	ℓ	N_{DD}	r_∞^ℓ	CPU
20	8	626	2.1e-3	713.5	200	7	492	6.5e-2	32.4
40	9	500	9.0e-3	619.8	400	5	404	1.6e-2	114.7
60	14	659	2.0e-2	769.9	600	6	464	7.1e-3	335.8
80	37	1313	3.9e-2	1522.4	800	6	458	4.0e-3	658.4

Table 1: Two–grid algorithm.

3 Numerical results

In this section, we report numerical performances of algorithm (5) with (5.iii) being solved by (12) on the coarse grid mesh. We choose the domain $\Omega = (0,1)^2$ and the coarse gird mesh size $H = \gamma h$, where $\gamma = 1, \cdots, 4$. When $\gamma = 1$, (5) is reduced to (12). The algorithm is implemented in FORTRAN and run on the SGI Power Challenge L with a 75MHz T8000 processor (a serial computer). The problem coefficients are given as $p(x,y) = \dfrac{\omega}{c(x,y)}$, $\alpha(x,y) = \omega$, $q = 0$, where freq is the frequency, $\omega = 2\pi \cdot \texttt{freq}$ denotes the angular frequency, and the wave speed

$$c(x,y) = \begin{cases} 3, & x \le 0.5, \\ 1 + e^x + \sin(2\pi xy), & x > 0.5. \end{cases} \tag{14}$$

The source function $f(x,y)$ is selected so that the true solution $u(x,y) = \dfrac{\phi(x)\cdot\phi(y)}{\omega^2}$, where $\phi(x) = e^{i\omega(x-1)} + e^{-i\omega x} - 2$. The domain is decomposed into M_x (resp., M_y) subdomains in x–(resp., y)–direction, respectively. The computation time is denoted by CPU (seconds). For the smoother S, we employ γ iterations of the symmetric Gauss Seidel (SGS) algorithm. The two–grid iteration (V–cycle) is stopped when the relative residual is less than 10^{-5} in the maximum norm and the inner loop solving (5.iii) is stopped with relative L^∞ error being smaller than 10^{-2}. We choose the parameters arising in the IADT (12) as

$$\eta = \frac{3}{4} \cdot \texttt{freq}, \quad n_* = \frac{M_x + M_y}{5}. \tag{15}$$

(See (13).) The number of V–cycles is denoted by ℓ and the total number of DD iterations by N_{DD}. The error is checked by the relative maximum norm $r_\infty^\ell = \dfrac{\|\mathbf{u}^{h,\ell} - u\|_\infty}{\|u\|_\infty}$, where $\mathbf{u}^{h,\ell}$ is the ℓ–th iterate of algorithm (5). Zero initial values are given: $\mathbf{u}^{h,0} \equiv 0$.

From various numerical experiences, we observed the following: The algorithm converges 3 times faster (measured in the computation time) than (12) when $H = 2h$ and 7–9 times faster when $H = 4h$. Such two–grid algorithms take benefits not only on the computation time but on the computer memory.

In Table 1, we check how the rate of convergence of the algorithm depends on the frequency and the mesh size h. In the first panel, we choose $1/h = 1080$, $H = 3h$, and $M_x \times M_y = 60 \times 1$, while we set $\texttt{freq} = 20$, $H = 2h$, and $M_x \times M_y = 50 \times 1$ for the second panel. To check dependence of the algorithm on wave number (resp., mesh size), four different frequencies (resp., mesh sizes) are selected in the first (resp., second) panel. It seems that the frequency effects on the number

of iterations *sub-linearly* and on the CPU time *little*, provided that the coarse grid problem is fine enough, i.e. wavelength $\geq 6H \sim 8H$. (When freq = 80, in the first panel, the wavelength is $4.5H$; we can hardly expect that the coarse grid problem captures characteristics of the physical problems; the algorithm converges even though it is *slow*. Note that the two-grid algorithm with the coarse grid mesh badly assigned is still worthwhile to use for solving the problem.) On the other hand, the number of iterations seems not affected by the mesh size h. So, the rate of convergence depends only on the wave number *sub-linearly* and *not* on the mesh size. From such a dependence, we can easily see that (5) is applicable to realistic/larger problems efficiently.

References

[1] A. Bayliss, C. Goldstein, and E. Turkel, *An iterative method for the Helmholtz equation*, J. Comput. Phys. 49 (1983), pp. 443–457.

[2] ———, *On accuracy conditions for the numerical computation of waves*, J. Comput. Phys. 59 (1985), pp. 396–404.

[3] B. Després, *Domain decomposition method and the Helmholtz problem*, in Mathematical and Numerical Aspects of Wave Propagation Phenomena, G. Cohen, L. Halpern and P. Joly, eds., SIAM, Philadelphia (1991), pp. 44–51.

[4] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal. 21 (1984), pp. 352–362.

[5] R. W. Freund, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM J. Sci. Stat. Comput. 13 (1992), pp. 425–448.

[6] W.D. Joubert and D.M. Young, *Necessary and sufficient conditions for the simplification of the generalized conjugate-gradient algorithms*, Linear Algebra Appl. 88/89 (1987), pp. 449–485.

[7] S. Kim, *Parallel multidomain iterative algorithms for the Helmholtz wave equation*, Appl. Numer. Math. 17 (1995), pp. 411–429.

[8] ———, *On the use of rational iterations and domain decomposition methods for solving the Helmholtz problem*, submitted.

[9] ———, *Domain decomposition iterative procedures for solving scalar waves in the frequency domain*, submitted.

[10] S. Kim and W.W. Symes, *Cell-centered finite difference modeling for the 3-D Helmholtz problem*, submitted.

[11] P. L. Lions, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, in Third International Symposium on Domain Decomposition Method for Partial Differential Equations, T.F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds., SIAM, Philadelphia (1990), pp. 202–223.

Multigrid for the Galerkin Least Squares Method in Linear Elasticity : The Pure Displacement Problem

Jaechil Yoo
Department of Mathematics
University of Wisconsin-Madison

December 15, 1995

Abstract.

In [5], Franca and Stenberg developed several Galerkin least squares methods for the solution of the problem of linear elasticity. That work concerned itself only with the error estimates of the method. It did not address the related problem of finding effective methods for the solution of the associated linear systems. In this work, we prove the convergence of a multigrid(W-cycle) method. This multigrid is robust in that the convergence is uniform as the parameter, ν, goes to $\frac{1}{2}$. Computational experiments are included.

1 Introduction

Let Ω be a bounded convex polygonal domain in R^2 and $\partial\Omega$ be the boundary of Ω. The pure displacement boundary value problem for planar linear elasticity is given in the form

$$2\mu\{\nabla\cdot\varepsilon(u)+\frac{\nu}{1-2\nu}\nabla\nabla\cdot u\}+f=0 \text{ in } \Omega, \quad u=0 \text{ on } \partial\Omega. \tag{1}$$

Here $u=(u_1,u_2)$ denotes the displacement, $f=(f_1,f_2)$ is the body force, ν is the Poisson's ratio and μ is the shear modulus given by $\mu=\frac{E}{2(1+\nu)}$ where E is the Young's modulus.

We restrict Poisson's ratio to $0\le\nu<\frac{1}{2}$ where the upper limit corresponds to an incompressible material. The explanation for the notations

used in (1) is given in [2] and [5].

It is well-known that one way of driving stabilized mixed finite element methods is to combine the classical Galerkin formulation with least-squares forms of the differential equations. (See [5] and references therein). An advantage of this method is that the class of finite element spaces that can be used are considerablely enlarged, hence the methods are easily incorporated into existing finite element codes. In [5], Franca and Stenberg developed several Galerkin least squares methods for the elasticity equations and proved the error estimates of their methods with the stabilization parameter α bounded by C_I, where C_I is the constant in the inverse inequality. But α and C_I are unknown. So, for the implementation of stabilized mixed finite element methods, we have to analyze the behaviors of α and C_I in order to obtain rapid iterative convergence.

As documented in [7], the standard multigrid method using conforming bilinear finite elements requires a large number of smoothing steps in order to achieve convergence for nearly incompressible linear elasticity problems. Our algorithm converges with a small number of smoothing steps and is the first multigrid algorithm in the implementation that uses P-1 finite element spaces for approximating both the displacement and the pressure.

In this paper, we develope a W-cycle multigrid method to solve the linear system arising from P-1 conforming finite element method for the mixed formulation of the pure displacement boundary value problem as in [2] and [6]. We prove the convergence of a W-cycle multigrid method as in [2] and [8]. We show that the convergence is uniform with respect to the parameter ν. We demonstrate that the number of iterations for our algorithm depends on the stabilization parameter α. Finally, we find the appropriate value of α for several cases.

2 The Finite Element Method

For simplicity, we assume that $2\mu = 1$. Let $p = -\frac{1}{\epsilon}\nabla \cdot u$, where $\epsilon = \frac{1-2\nu}{\nu}$. Then (1) is equivalent to

$$
\begin{aligned}
&-\nabla \cdot \varepsilon(u) + \nabla p = f \ \text{ in } \Omega \\
&\epsilon p + \nabla \cdot u = 0 \ \text{ in } \Omega \\
&u = 0 \ \text{ on } \partial\Omega
\end{aligned}
\tag{2}
$$

Hence, we have the following weak formulation:

Find $(u,p) \in H_0^1(\Omega) \times L^2(\Omega)$ such that

$$(3) \qquad \begin{aligned} (\nabla \cdot \varepsilon(u), \nabla \cdot \varepsilon(v)) - (\nabla \cdot v, p) &= (f,v), \quad \forall v \in H_0^1(\Omega), \\ \epsilon(p,q) + (\nabla \cdot u, q) &= 0 \quad \forall q \in L^2(\Omega). \end{aligned}$$

With the first differential equation in (2), it is clear that (3) is equvalent to the following stabilized mixed formulation:

Find $(u,p) \in H_0^1(\Omega) \times L^2(\Omega)$ such that

$$(4) \qquad \mathcal{B}((u,p),(v,q)) = \mathcal{F}_f(v,q) \quad \forall (v,w) \in H_0^1(\Omega) \times L^2(\Omega).$$

where

$$\begin{aligned} \mathcal{B}((u,p),(v,q)) &= (\varepsilon(u) : \varepsilon(v)) - (\nabla \cdot u, q) - (\nabla \cdot v, p) \\ &\quad -\alpha \sum_{T \in \mathcal{T}^k} h_T^2 (-\nabla \cdot \varepsilon(u) + \nabla p, -\nabla \cdot \varepsilon(v) + \nabla q)_T \\ &\quad -\epsilon(p,q) \end{aligned}$$

and

$$\mathcal{F}_f(v,q) = (f,v) - \alpha \sum_{T \in \mathcal{T}^k} h_T^2 (f, -\nabla \cdot \varepsilon(v) + \nabla q)_T$$

Let $\mathcal{T}^k$ be a family of triangulations of Ω, where $\mathcal{T}^{k+1}$ be obtained by connecting the midpoints of the edges of the triangles in $\mathcal{T}^k$. Let $h_T = diam T$ for each $T \in \mathcal{T}^k$ and $h_k = max_{T \in \mathcal{T}^k} h_T$, then $h_k = 2h_{k+1}$. Now let's define the conforming finite element spaces for our multigrid method.

$$\begin{aligned} V_k &:= \{v \in C^0(\Omega); v|_T \in \mathcal{P}_1(T), \forall T \in \mathcal{T}^k\} \\ P_k &:= \{q \in C^0(\Omega); q|_T \in \mathcal{P}_1(T), \forall T \in \mathcal{T}^k\} \ \textit{and} \\ \tilde{P}_k &:= \{q \in C^0(\Omega); q|_T \in \mathcal{P}_1(T), \int_\Omega g dx = 0, \forall T \in \mathcal{T}^k\} \end{aligned}$$

Then the discretized problem for (4) is the following :

Find $(u_k, p_k) \in V_k \times \tilde{P}_k$ such that

$$(5) \qquad \mathcal{B}_k((u_k,p_k),(v,q)) = \mathcal{F}_f(v,q) \text{ for all } (v,q) \in V_k \times \tilde{P}_k$$

where

$$\begin{aligned} \mathcal{B}_k((u_k,p_k),(v,q)) &= (\varepsilon(u_k) : \varepsilon(v)) - (\nabla \cdot u_k, q) - (\nabla \cdot v, p_k) \\ &\quad -\alpha \sum_{T \in \mathcal{T}^k} h_T^2 (-\nabla \cdot \varepsilon(u_k) + \nabla p_k, -\nabla \cdot \varepsilon(v) + \nabla q)_T \\ &\quad -\epsilon(p_k,q) \end{aligned}$$

and

$$\mathcal{F}_f(v,q) = (f,v) - \alpha \sum_{T \in T^k} h_T^2 (f, -\nabla \cdot \varepsilon(v) + \nabla q)_T$$

Note that the bilinear form $\mathcal{B}_k$ is symmetric and indefinite.

In [5], Franca and Stenberg proved the uniqueness of the solution of the conforming discretization (5) and derived the following discretization error estimate :

$$|u - u_k|_{H^1(\Omega)} + |p - p_k|_{L^2(\Omega)} \leq C h_k |f|_{L^2(\Omega)}$$

for $0 < \alpha < C_I$ where C_I is the constant satisfying the inverse inequality,

$$C_I \sum_{T \in T^k} h_T^2 |\nabla \cdot \varepsilon(v)|_T^2 \leq |\varepsilon(v)|_{L^2(\Omega)}^2, \quad \forall v \in V_k.$$

3 Multigrid Algorithm

In this section we present lemmas and theorems without proofs which are found in [9].

In order to define the fine-to-coarse operator I_k^{k-1}, we introduce the following mesh-dependent inner product:

$$((u,p),(v,q))_k := (u,v)_{L^2(\Omega)} + h_k^2 (p,q)_{L^2(\Omega)}.$$

Then $I_k^{k-1} : V_k \times P_k \to V_{k-1} \times P_{k-1}$ is defined by

$$(I_k^{k-1}(u,p),(v,q))_{k-1} = ((u,p),(v,q))_k$$

for all $(u,p) \in V_k \times P_k$ and $(v,q) \in V_{k-1} \times P_{k-1}$.

Define $B_k : V_k \times P_k \to V_k \times P_k$ by

$$(B_k(u,p),(v,q))_k = \mathcal{B}_k((u,p),(v,q)),$$

for all $(u,p),\ (v,q) \in V_k \times P_k$.

Lemma 1 *(i) Given* $(u,p) \in V_k \times P_k$,

$$(u,p) \in V_k \times \tilde{P}_k \Leftrightarrow ((u,p),(0,1))_k = 0.$$

(ii) $I_k^{k-1} : V_k \times \tilde{P}_k \to V_{k-1} \times \tilde{P}_{k-1}$.

Lemma 2 *The subspace* $V_k \times \tilde{P}_k$ *is invariant under* B_k.

Let $\tilde{B}_k = B_k \mid_{V_k \times \tilde{P}_k}$.

Lemma 3 *Spectral radius of* $\tilde{B}_k \leq Ch_k^{-2}$.

The mesh-dependent norms on $V_k \times \tilde{P}_k$ are defined as follows

$$|||(u,p)|||_{s,k} := \sqrt{((\tilde{B}_k^2(u,p),(u,p))_k} \text{ for all } (u,p) \in V_k \times \tilde{P}_k$$

Note that

$$|||(u,p)|||_{0,k} := \sqrt{||u||_0^2 + h_k^2||p||_0^2} \text{ for all } (u,p) \in V_k \times \tilde{P}_k$$

Let

$$\begin{aligned} \mathcal{B}_{k-1}^*((u,p),(v,q)) &= (\varepsilon(u):\varepsilon(v)) - (\nabla \cdot u, q) - (\nabla \cdot v, p) \\ &\quad -\alpha/4 \sum_{T \in T^{k-1}} h_T^2(-\nabla \cdot \varepsilon(u) + \nabla p, -\nabla \cdot \varepsilon(v) + \nabla q)_T \\ &\quad -\epsilon(p,q) \end{aligned}$$

and

$$\mathcal{F}_f^*(v,q) = (f,v) - \alpha/4 \sum_{T \in T^{k-1}} h_T^2(f, -\nabla \cdot \varepsilon(v) + \nabla q)_T$$

Define $P_k^{k-1} : V_k \times \tilde{P}_k \to V_{k-1} \times \tilde{P}_{k-1}$ by

$$\mathcal{B}_{k-1}^*(P_k^{k-1}(u,p),(v,q)) = \mathcal{B}_k((u,p),(v,q))$$

for all $(u,p) \in V_k \times \tilde{P}_k$ and $(v,q) \in V_{k-1} \times \tilde{P}_{k-1}$

We describe the k-th level iteration scheme of the conforming multigrid algorithm. The k-th level iteration with initial guess (y_0, z_0) yields $CMG(k,(y_0,z_0),(w,r))$ as a conforming approximate solution to the following problem.

Find $(y,z) \in V_k \times \tilde{P}_k$ such that

$$\tilde{B}_k(y,z) = (w,r), \text{ where } (w,r) \in V_k \times \tilde{P}_k$$

For $k = 1$, $CMG(1,(y_0,z_0),(w,r))$ is the solution obtained from a direct method. In other words,

$$CMG(1,(y_0,z_0),(w,r)) = (\tilde{B}_1)^{-1}(w,r)$$

For $k > 1$, there are two steps.
Smoothingstep : Let $(y_m, z_m) \in V_k \times P_k$ be defined recursively by the initial guess (y_0, z_0) and the equations

$$(y_l, z_l) = (y_{l-1}, z_{l-1}) + \frac{1}{\Lambda_k^2} B_k((w, r) - B_k(y_{l-1}, z_{l-1})), \quad 1 \leq l \leq m$$

where $\Lambda_k := Ch_k^{-2}$ is greater than or equal to the spectral radius of $\tilde{B}_k$, and m is a positive integer to be determined later.
Correctionstep : The coarser-grid correction in $V_k \times P_k$ is obtained by applying the $(k-1)$-th level conforming iteration twice. More precisely,

$$(v_0, q_0) = (0, 0) \text{ and}$$
$$(v_i, q_i) = CMG(k-1, (v_{i-1}, q_{i-1}, (\bar{w}, \bar{r}), \quad i = 1, 2$$

where $(\bar{w}, \bar{r}) \in V_{k-1} \times P_{k-1}$ is defined by $(\bar{w}, \bar{r}) := I_k^{k-1}((w, r) - B_k(y_m, z_m))$. Then $CMG(k, (y_0, z_0), (w, r)) = (y_m, z_m) + I_{k-1}^k(v_2, q_2)$.

Remark. In the smoothing step, we use B_k instead of $\tilde{B}_k$. Because the space $V_k \times P_k$ has a natural coordinate system which consists of the values of piecewise linear functions at mesh points on the triangles. In view of Lemma 1 and Lemma 2, the result of the smoothing step and the correction step belongs to $V_k \times \tilde{P}_k$. Therefore, in the actual implementation of the multigrid method, we use only the natural coordinate system of $V_k \times P_k$. Note that B_k is represented by a sparse banded matrix and B_k is not invertible.

Now we discuss the convergence of the two-grid algorithm where the residual equation is solved exactly on the coarser grid. Let the final output of the two-grid algorithm be

$$(y^*, q^*) := (y_m, z_m) + (v^*, q^*)$$

where $(v^*, q^*) = (\tilde{B}_{k-1})^{-1} I_k^{k-1} B_k(y - y_m, z - z_m)$.

Lemma 4

$$(v^*, q^*) = P_k^{k-1}(y - y_m, z - z_m).$$

Let the k-th level relaxation operator R_k be defined by

$$R_k := I - \frac{1}{\Lambda_k^2}(B_k)^2.$$

Then we have

$$(y - y_m, z - z_m) = R_k^m(y - y_0, z - z_0)$$
$$(y - y^*, z - z^*) = (I - P_k^{k-1})R_k^m(y - y_0, z - z_0).$$

Lemma 5 Smoothing Step *There exists a constant C, independent of h_k and m, such that*

$$\|R_k^m(u,p)\|_{2,k} \leq Ch_k^{-2}\frac{1}{\sqrt{m}}\|(u,p)\|_{0,k}, \quad \forall (u,p) \in V_k \times \tilde{P}_k$$

Lemma 6 Approximation Step *There exists a constant C, independent of h_k and m, such that*

$$\|(I - P_k^{k-1})(u,p)\|_{0,k} \leq Ch_k^2\|(u,p)\|_{2,k}, \quad \forall (u,p) \in V_k \times \tilde{P}_k$$

Theorem 1 Convergence of the Two-Grid Algorithm *There exists a constant C, independent of k and m, such that*

$$\|(y - y^*, z - z^*)\|_{0,k} \leq \frac{C}{\sqrt{m}}\|(y - y_0, z - z_0)\|_{0,k}.$$

Theorem 2 Convergence of the k-th Level Algorithm *There exists a constant C, independent of k and m, such that*

$$\|(y,z) - CMG(k,(y_0,z_0),(w,r))\|_{0,k} \leq \frac{C}{\sqrt{m}}\|(y - y_0, z - z_0)\|_{0,k}.$$

4 Experimental Results

We apply the V-cycle and W-cycle multigrid algorithm to the pure displacement boundary value problem (1) studied in [2]. The domain Ω is the unit square, and the body force $f = (f_1, f_2)$ is taken to be as follows :

$$f_1 = \pi^2[2\sin 2\pi y(-1 + 2\cos 2\pi x) - 0.5\cos\pi(x+y) + \frac{\epsilon}{\epsilon+2}\sin\pi x \sin\pi y],$$
$$f_2 = \pi^2[2\sin 2\pi x(1 - 2\cos 2\pi y) - 0.5\cos\pi(x+y) + \frac{\epsilon}{\epsilon+2}\sin\pi x \sin\pi y].$$

The exact solution $u = (u_1, u_2)$ is

$$u_1 = \sin 2\pi y(-1 + \cos 2\pi x) + \frac{\epsilon}{\epsilon+2}\sin\pi x \sin\pi y,$$
$$u_2 = \sin 2\pi x(1 - \cos 2\pi y) + \frac{\epsilon}{\epsilon+2}\sin\pi x \sin\pi y.$$

	$V-CYCLE$				$W-CYCLE$			
smoothing	α=3	α=1	α=0.1	α=0.01	α=3	α=1	α=0.1	α=0.01
1	1190	937	840	831	446	327	284	280
2	595	469	420	416	224	164	142	140
3	397	313	281	278	149	110	95	94
4	298	235	211	208	112	82	71	70

Table 1: Number of grid = 32 i.e. h = 1/32 and $\nu = 0.3$

	$V-CYCLE$				$W-CYCLE$			
smoothing	α=3	α=1	α=0.1	α=0.01	α=3	α=1	α=0.1	α=0.01
1	1106	839	766	762	437	304	270	268
2	553	420	383	381	219	152	135	135
3	369	280	256	254	146	102	90	90
4	277	210	192	191	110	76	68	68

Table 2: Number of grid = 32 i.e. h = 1/32 and $\nu = 0.45$

	$V-CYCLE$				$W-CYCLE$			
smoothing	α=3	α=1	α=0.1	α=0.01	α=3	α=1	α=0.1	α=0.01
1	1081	825	774	773	437	296	271	271
2	541	413	387	387	219	148	136	136
3	361	276	259	258	146	99	91	91
4	271	207	194	194	110	75	68	68

Table 3: Number of grid = 32 i.e. h = 1/32 and $\nu = 0.495$

	$V-CYCLE$				$W-CYCLE$			
smoothing	α=5	α=3	α=1	α=0.1	α=5	α=3	α=1	α=0.1
1	1501	1079	826	777	659	437	296	272
2	754	540	413	diverge	330	219	148	diverge
3	503	360	diverge	diverge	220	146	diverge	diverge
4	378	diverge	diverge	diverge	165	110	diverge	diverge

Table 4: Number of grid = 32 i.e. h = 1/32 and $\nu = 0.4995$

We follow the implementation of our algorithm as in [6]. Let the number of grid be 32, i.e. the mesh size $h = \frac{1}{32}$. The programs excute until the discrete L^2 relative error is less than 5% of the initial error. We use the initial guesses, $u^0 = (u_1^0, u_2^0) = (0,0)$ and $p^0 = 0$. The computations were done in double-precision arithmetic for various α, smoothing steps and ν, where $\nu = \frac{\gamma}{2(\gamma+1)}$ is the Poisson ratio.

First note that, in practice, our algorithm converges even for the V-cycle with one smoothing step for appropriate α's, while we have only proven the convergence of W-cycle with sufficiently many smoothing steps.
The numbers in columns represent the number of iterations to achieve a L^2 relative error of less than 5% in the displacement.

In table 1,2 and 3, we get an appropriate α for each case.
For the V-cycle, $\alpha = 0.01$ is the appropariate value for the case of $\nu =$ 0.3, 0.45 and 0.495. For the W-cycle, $\alpha = 0.1$ (or 0.01) is the appropriate value.
In table 4, case of $\nu = 0.4995$, we have to choose relatively big α in order that our algorithm converges.

Acknowledgement. I would like to thank Professor Seymour V. Parter for his advice and encouragement.

References

[1] R.E.BANK AND T.DUPONT; "An optimal order process for solving finite element equations", Math. Comp., **36**, 827-835, (1981).

[2] S.C.BRENNER; "A nonconforming mixed multigrid method for the pure displacement problem in planar linear elasticity", SIAM J. Numer. Anal., **30**, 116-135, (1993).

[3] F.BREZZI AND J.DOUGLAS, Jr.; "Stabilized mixed methods for the Stokes problem" Numer. Math., **53**, 225-235, (1988).

[4] P. CIARLET; "The finite element method for elliptic problems", North-Holland, Amsterdam, (1978).

[5] L.P.FRANCA AND R.STENBERG; "Error analysis of some Galerkin least squares methods for the elasticity equations", SIAM J. Numer. Anal., **28**, 1680-1697, (1991).

[6] C.-O.LEE; "Multigrid methods for the pure traction problem of linear elasticity : mixed formulation", preprint.

[7] I.D.PARSONS AND J.F.HALL; "The multigrid method in solid mechanics : Part I -algorithm description and behavior", Internat. J. Meth. Engrg., **29**, 719-738, (1990).

[8] R.VERFURTH; "A multilevel algirithm for mixed problems", SIAM J. Numer. Anal., **21**, 264-271, (1984).

[9] J.YOO "Multigrid method for the Galerkin least squares method in linear elasticity : The pure displacement problem", preprint.