NASA Conference Publication 3339
Part 2

Seventh Copper Mountain Conference on Multigrid Methods

Edited by
N. Duane Melson, Tom A. Manteuffel, Steve F. McCormick, and Craig C. Douglas

Proceedings of a workshop cosponsored by the
National Aeronautics and Space Administration,
Washington, D.C., and the Department of Energy,
Washington, D.C., and held at
Copper Mountain, Colorado
April 2–7, 1995

September 1996

NASA Conference Publication 3339
Part 2

Seventh Copper Mountain Conference on Multigrid Methods

Edited by
N. Duane Melson
Langley Research Center • Hampton, Virginia

Tom A. Manteuffel and Steve F. McCormick
University of Colorado • Boulder, Colorado

Craig C. Douglas
IBM Thomas J. Watson Research Center • Yorktown Heights, New York
Yale University • New Haven, Connecticut

Proceedings of a workshop cosponsored by the
National Aeronautics and Space Administration,
Washington, D.C., and the Department of Energy,
Washington, D.C., and held at
Copper Mountain, Colorado
April 2–7, 1995

National Aeronautics and Space Administration
Langley Research Center • Hampton, Virginia 23681-0001

September 1996

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

This publication is available from the following sources:

NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

PREFACE

The *Seventh Copper Mountain Conference on Multigrid Methods* was held on April 2–7, 1995, at Copper Mountain, Colorado, and was sponsored by NASA and the Department of Energy. The University of Colorado, Front Range Scientific Computations, Inc., and the Society for Industrial and Applied Mathematics provided organizational support for the conference.

This document is a collection of many of the papers that were presented at the conference and thus represents the conference proceedings. NASA Langley has graciously provided printing of this book so that all of the papers could be presented in a single forum. Each paper was reviewed by a member of the conference organizing committee under the coordination of the editors.

The multigrid discipline continues to expand and mature, as is evident from these proceedings. The vibrancy and diversity in this field are amply expressed in these important papers, and the collection clearly shows the continuing rapid growth of the use of multigrid acceleration techniques.

N. Duane Melson
NASA Langley Research Center

Steve F. McCormick and
Tom A. Manteuffel
University of Colorado at Boulder

Craig Douglas
IBM Thomas J. Watson Research Center
Yale University

The use of trademarks or names of manufacturers in this publication does not constitute endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

ORGANIZING COMMITTEE

Joel Dendy
Los Alamos National Laboratory

Craig Douglas
IBM/Yale University

Paul Frederickson
RIACS

Van Henson
Naval Postgraduate School

Jan Mandel
University of Colorado at Denver

Tom Manteuffel
University of Colorado

Steve McCormick
University of Colorado

Duane Melson
NASA Langley Research Center

Seymour Parter
University of Wisconsin

Joseph Pasciak
Brookhaven National Laboratory

John Ruge
University of Colorado at Denver

Klaus Stueben
Gesellschaft f. Math. u. Datenverarbeitung

Pieter Wesseling
Delft University

Olof Widlund
Courant Institute

ATTENDEES

LOYCE ADAMS	adams@amath.washington.edu
FERNANDO ALVARADO	alvarado@engr.wisc.edu
EYAL ARIAN	arian@icase.edu
STEVE ASHBY	sfashby@llnl.gov
VICTOR BANDY	vab@swan.lanl.gov
DANA BEDIVAN	bedivan@utamat.uat.edu
M. BERNDT	berndt@colorado.edu
PAVEL BOCHEV	bochev@utamat.uta.edu
JAMES BORDNER	bordner@cs.uiuc.edu
A. BORZI	Alfio.Borzi@comlab.ox.ac.uk
ACHI BRANDT	mabrandt@weizmann.weizmann.ac.il
SUSANNE BRENNER	brenner@math.scarolina.edu
MARIAN BREZINA	mbrezina@tiger.cudenver.edu
OLIVER BROKER	broker@cs.colorado.edu
JAN BROEZE	j.broeze@math.utwente.nl
ZHIQIANG CAI	zcai%mathd.usc.edu@usc.edu
XIAO-CHUAN CAI	cai@schwarz.cs.colorado.edu
PHIL CALVIN	mudpuppy@gibbs.oit.unc.edu
DAVID CANRIGHT	dcanright@nps.navy.mil
MARIO CASARIN	casarin@math1.nyu.edu
RICHARD CASEY	richard.casey@asu.edu
ZHANGXIN CHEN	zchen@golem.math.smu.edu
REGINALD W. CLEMENS	reg@dwf.com (or) clemens@plk.af.mil
A. W. CRAIG	Alan.Craig@sima.sintef.no
GENE D'YAKANOV	dknv@cmc.msk.su
BRUCE DAVIS	davis@cfdlab.ae.utexas.edu
JOEL DENDY	jed@lanl.gov
QINGPING DENG	deng@math.utk.edu
CRAIG DOUGLAS	douglas-craig@cs.yale.edu
JON DYM	jdym@cams.usc.edu

HOWARD ELMAN	elman@cs.umd.edu
BJORN ENGQUIST	engquist@math.ucla.edu
R. D. FALGOUT	falgout@bacchus.llnl.gov
CHARBEL FARHAT	charbel@alexandra.colorado.edu
HERMANN FASEL	faselh@ccit.arizona.edu
JEAN MICHEL FIARD	fiard@newton.colorado.edu
PAUL FREDERICKSON	MathCube@aol.com
KLAUS GARTNER	gaertner@iis.ee.ethz.ch
THOR GJESDAL	thor@cmr.no
SIMON GLEYZER	gleyzer@gibbs.oit.unc.edu
WOJCIECH GOLIK	golik@arch.umsl.edu
HERVE GUILLARD	Herve.Guillard@inria.fr
VAN HENSON	vhenson@boris.math.nps.navy.mil
ALAN HEROD	aherod@newton.colorado.edu
GREGORY HILL	ghill@cs.colorado.edu
LOUIS HOWELL	nazgul@bigbird.llnl.gov
JEROME JAFFRE	Jerome.Jaffre@inria.fr
JIM JONES	jijones@mtha.usc.edu
KIRK JORDAN	kjordan@vnet.ibm.com
MICHAEL JUNG	Dr.Michael.Jung@mathematik.tu-chemnitz
DAVID KINCAID	kincaid@cs.utexas.edu
AXEL KLAWONN	klawonn@goedel.uni-muenster.de
ANDREW KNYAZEV	aknyazev@tiger.cudenver.edu
HWAR-CHING KU	ku@aplcomm.jhuapl.edu
CHEN-YAO G. LAI	cylai@math.ccu.edu.tw
R. LAZAROV	lazarov@math.tamu.edu
CHANG OCK LEE	colee@math.inha.ac.kr
BARRY LEE	blee@boulder.colorado.edu
G. SCOTT LETT	slett@ssii.com
YONG LI	lyong@digger.gsfc.nasa.gov
C. LIU	cliu@carbon.denver.colorado.edu
ZHINING LIU	zliu@evans.denver.colorado.edu

SERGUEI MALIASSOV	malyasov@isc.tamu.edu
JAN MANDEL	jmandel@carbon.denver.colorado.edu
TOM MANTEUFFEL	tmanteuf@newton.colorado.edu
TAREK MATHEW	mathew@ledaig.uwyo.edu
STEVE MCCORMICK	stevem@newton.colorado.edu
S. MCKAY	mckay@math.byu.edu
ROBERT MCLAY	mclay@cfdlab.ae.utexas.edu
A .J. MEIR	ajm@math.auburn.edu
DUANE MELSON	n.d.melson@larc.nasa.gov
ILYA MISHEV	mishev@isc.tamu.edu
WILLIAM MITCHELL	mitchell@cam.nist.gov
HANS MOLENAAR	hansmo@twi.tudelft.nl
SERGEI NEPOMNYASCHIKH	svnep@comcen.nsk.su
JOHN W. NEUBERGER	jwn@unt.edu
ELYAS NURGAT	nurgat@scs.leeds.ac.uk
SUELY OLIVEIRA	suely@cs.tamu.edu
MARY OMAN	imsgmoma@math.montana.edu
MARIA ELIZABETH ONG	ong@sdna5.ucsd.edu
ROSSEN PARASHKEVOV	rossen@newton.colorado.edu
SEYMOUR PARTER	parter@cs.wisc.edu
JOE PASCIAK	pasciak@bnl.gov
JAN PEETERSWEEM	peetersw@newton.colorado.edu
CHRISTOPH PFLAUM	pflaum@informatik.tu-muenchen.de
J. R. PHILLIPS	jphill@rle-vlsi.mit.edu
KLAUS RESSEL	kressel@tiger.cudenver.edu
KRIS RIEMSLAGH	Kris.Riemslagh@rug.ac.be
GUY ROBINSON	robinson@npac.syr.edu
JOHN RUGE	jruge@boulder.colorado.edu
TORGEIR RUSTEN	Torgeir.Rusten@si.sintef.no
FAISAL SAIED	saied@cs.uiuc.edu
MARKUS SARKIS	msarkis@tigger.cs.colorado.edu
AIHUA SHAKER	ashaker@afit.af.mil

YAIR SHAPIRA	yair@csc.cs.technion.ac.il
DAVID SIDILKOVER	sidilkov@icase.edu
PETER STAAB	staab@newton.colorado.edu
GERHARD STARKE	starke@boulder.colorado.edu
ANDREAS STATHOPOULOS	andreas@vuse.vanderbilt.edu
WILLIAM J. STEWART	
DANIEL B. SZYLD	szyld@euclid.math.temple.edu
RADEK TEZAUR	tezaur@tiger.cudenver.edu
MARIETTA TRETTER	eo21mt@tamvm1.tamu.edu
ALEXANDER TROFIMOV	fmm@uni.tiv.dnepropetrovsk.ua
STEFAN VANDEWALLE	stefan@ama.caltech.edu
PETR VANEK	pvanek@tiger.cudenver.edu
PRATAP VANKA	vanka@uy.ncsa.uiuc.edu
APOSTOL VASSILEV	apostol@isc.tamu.edu
C. VUIK	cvuik@math.tudelft.nl
HONG WANG	hwang@math.scarolina.edu
JUNPING WANG	junping@schwarz.uwyo.edu
RUIKE WANG	wang@rsci.ssii.com
OLOF WIDLUND	widlund@widlund.cs.nyu.edu
STEPHEN B. WINEBERG	wineberg@math.lsa.umich.edu
KRISTIAN WITSCH	witsch@numerik.uni-duesseldorf.de
DEXUAN XIE	xie@math.uh.edu
JINCHAO XU	xu@math.psu.edu
IRAD YAVNEH	irad@cs.technion.ac.il
DAVID YOUNG	young@cs.utexas.edu
XIUYANG YU	xyu@carbon.denver.colorado.edu
LEONID ZASLAVSKY	zasl@wisdom.weizmann.ac.il
X. ZHENG	xzheng@tiger.cudenver.edu

MULTIGRID HISTORY

(At the awards ceremony of the conference, Achi Brandt presented the following history of multigrid. The reader should study the truths contained herein and revel in the humor.)

The early history of multigrid has recently become a hot subject of research. An ancient multigrid code was uncovered during extensive excavations last year in northern Turkestan. Carbon tests indicate that this code has an efficiency of 5.1 on the Richter scale. Some researchers believe that the V cycle was practiced by the Neanderthals. The use of the Full Multigrid (FMG) algorithm was, however, unique to Homo sapiens and is one of the major reasons for their ultimate survival. Prototypes of two-grid algorithms predate the first hominids. Most historians agree that coarsening was, in fact, invented by the dinosaurs; however, coarse-to-fine grid transfers were unknown to them, which explains their extinction.

Earlier geological findings include rich multilevel deposits that have been unearthed in several North American gold mines, and thick layers of old multigridders have been discovered at Copper Mountain.

The artifacts at the northern Turkestan site indicate that an early form of residual weighting was already in widespread use before the middle Full Approximation Storage (FAS) period. When Copernicus first introduced line relaxation, it was banned by the Catholic church. Pope Pointus the Square decreed that mere mortals should not practice such nonlocal schemes. He feared this practice would lead humanity to incompleteness, in particular to the incomplete LU decomposition of the Dutch church. The advent of variational coarsening during the French Revolution marks the dawn of the modern era, which is quite familiar to us all.

Page intentionally left blank

CONTENTS

Part 1*

*Part 1 is presented under separate cover.

Part 2

A PRESSURE BASED MULTIGRID PROCEDURE FOR THE NAVIER-STOKES EQUATIONS ON UNSTRUCTURED GRIDS

R. Jyotsna and S. P. Vanka
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign, Urbana, IL. 61801

ABSTRACT

We present details and performance of a pressure based multigrid solution procedure for the Navier-Stokes equations discretized on triangular grids. The discretization uses a control volume methodology, with linear inter-nodal variation of the flow variables. The use of the multigrid technique provides rapid and grid-independent rates of convergence. Three model driven cavity flows are computed, and the performance of the method at several grid densities and Reynolds numbers is reported. Representative flow fields characterizing the viscous eddies are also presented.

1. INTRODUCTION

The multigrid technique [1] provides an efficient means of smoothing high and low frequency errors that arise during the iterative solution of elliptic equations. Multigrid acceleration of solution procedures on unstructured meshes has been demonstrated earlier for single elliptic equations [2,3], for Euler equations [4-7], and for the compressible Navier-Stokes equations [8]. These procedures have used complete remeshing to generate a sequence of independent coarse and fine grids. Because of the independence of the grids, inter-grid transfers are somewhat complicated. Another strategy to coarsen a given fine grid is 'volume agglomeration', where the fine grid control volumes are progressively combined to obtain coarser control volumes. The resulting coarse grid volumes in this procedure do not have the same shapes as those of the finest grid, thus requiring special practices for constructing the discrete operators. The volume agglomeration technique is reviewed in reference [6].

The present paper describes a pressure based multigrid calculation procedure for unstructured grids. The discretization scheme is based on a control volume integration of the governing equations analogous to the practices followed in references [9-12]. On any given grid, the solution procedure employs a decoupled relaxation in conjunction with a pressure equation obtained through combination of the continuity and momentum equations in a special way [10]. In contrast with the coupled multigrid procedure followed in Vanka [13], and recently in Webster [14], the decoupled solution procedure is simpler to implement, and is better suited for use with a variety of linear solvers. In this paper, we discuss the details of the multigrid implementation, and its performance in three model driven-cavity flows. We have considered as examples, flows in a square cavity, a triangular cavity, and a semicircular cavity. The flow domain is discretized by Delaunay triangulation [15], with the fine grid obtained by uniform refinement of each triangle. In the following sections, we first describe the single grid procedure and its performance at increasing refinements of the mesh. Next, we describe the details of the components of the multigrid procedure (coarse grid equations, restriction, prolongation). The performance of the procedure in the three configurations at increasing Reynolds numbers is next presented along with brief descriptions of the flow fields.

2. GOVERNING EQUATIONS AND DISCRETIZATION PROCEDURE

Currently, we consider only the Navier-Stokes equations governing a two-dimensional, steady, incompressible flow of constant fluid properties. Thus the equations that are solved can be written in primitive variables (u, v, p) as

$$\nabla\cdot(\mathbf{u}\,u) = -(\partial p/\partial x) + \nu\,\nabla\cdot(\nabla u) + B_u \tag{1}$$

$$\nabla\cdot(\mathbf{u}\,v) = -(\partial p/\partial y) + \nu\,\nabla\cdot(\nabla v) + B_v \tag{2}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{3}$$

Here u and v are the two components of the velocity vector **u**, and p is the pressure divided by the density; ν is the kinematic viscosity, and B_u and B_v provide a means to include other forces such as those due to gravity and rotation.

The above equations are discretized on a triangular mesh shown in Figure 1(a). We use a control volume procedure essentially the same as that described in Prakash and Patankar [10], except that we have preferred to retain the central differencing scheme. In Prakash and Patankar [10] and related works, an exponential variation was introduced for stability at high cell Peclet numbers. Such a differencing scheme, although it provides stability, reduces the accuracy to first order, and is not satisfactory. Currently we have refined the finest mesh, until the cell Peclet number decreases below the stable value. Thus for a given grid, there exists a maximum flow Reynolds number that cannot be exceeded.

Figure 1(a) shows the control volume constructed around a representative node P, by joining the centroids of the relevant triangles to the midpoints of the sides. The equations are integrated over each of these control volumes to obtain nodal values of pressure and velocity. The checker-board split in the pressure field that arises in such equal-order interpolation is avoided, by requiring a different set of velocities ($\tilde{u}$, $\tilde{v}$), located at the cell interfaces, to satisfy mass continuity. This practice is similar to the momentum interpolation concept used in collocated finite volume schemes [16-18].

The Momentum Balances

Integrating equation (1) over the discrete control volume ABCDEF and using the divergence theorem, we have

$$\int_S [(\mathbf{u}\,u - \nu\nabla u)\cdot\mathbf{n}]\,dS = \int_V (B_u - \frac{\partial p}{\partial x})\,dV \tag{4}$$

where S is the enclosing surface of control volume V.

Consider now element PAB (Figure 1(b)), which has two faces a_1c and ca_3 bounding the control volume around P. The contributions from these two surfaces to the flux balance can be written as

$$\int_{a_1}^{c} (\mathbf{J}_u\cdot\mathbf{n})\,dS + \int_{c}^{a_3} (\mathbf{J}_u\cdot\mathbf{n})\,dS - \int_{Pa_1ca_3} (B_u - \frac{\partial p}{\partial x})\,dV \tag{5}$$

where $\qquad \mathbf{J}_u = \mathbf{u}\,u - \nu\nabla u$

To compute the flux $\mathbf{J}_u$, we use a linear interpolation of velocities between the nodes of PAB. Pressure is also assumed to vary linearly. Further, it is convenient to integrate the flux terms in local coordinates (X, Y), defined with the origin at the centroid of the element. The components of

J_u are then expressed in terms of the nodal values of u because of the linear interpolation used. Using Simpson's rule to evaluate the integrals, it can be shown that after collecting like terms and simplifying the complete equation, the resulting equation has the form

$$A_P u_P = \Sigma A_{nb} u_{nb} - V_P \langle B_u - \frac{\partial p}{\partial x} \rangle_P \tag{6}$$

where u_P is the value of u at point P and u_{nb} represents values at the neighboring nodes A, B, C, D, E and F. V_P is the area of the control volume around P, and ‹ › is an average defined by

$$\langle B \rangle = (1/V_P) \Sigma_e [(A_i / 3) B_i] \tag{7}$$

where A_i is the area of element i around P, and Σ_e denotes summation over all the elements contributing to V_P. The expressions for the coefficients are not provided here, but can be derived by the above mentioned steps. Following the same procedure for equation (2), we can obtain the discretized y-momentum balance as

$$A_P v_P = \Sigma A_{nb} v_{nb} - V_P \langle B_v - \frac{\partial p}{\partial y} \rangle_P \tag{8}$$

It is convenient to define momentum velocities $\hat{u}$ and $\hat{v}$ as

$$\hat{u} = (\Sigma A_{nb} u_{nb}) / A_P, \qquad \hat{v} = (\Sigma A_{nb} v_{nb}) / A_P \tag{9}$$

so that

$$u = \hat{u} + V_P \langle B_u - \frac{\partial p}{\partial x} \rangle / A_P \quad \text{and} \quad v = \hat{v} + V_P \langle B_v - \frac{\partial p}{\partial y} \rangle / A_P \tag{10}$$

The Continuity Equation

In the present procedure, u and v located at the nodal points do not satisfy the continuity equation. Rather, the cell face fluxes are balanced for each control volume. These cell face fluxes are interpolants of the nodal values in a special way that preserves the connections between the nodal pressures. The practice is similar to the momentum interpolation scheme used in finite volume schemes with a collocated arrangement of velocities and pressure [16-18].

We define a new set of velocities $\tilde{u}$ and $\tilde{v}$, located at the interfaces, and related to $\hat{u}$ and $\hat{v}$ by

$$\tilde{u} = \hat{u} + D (B_u - \frac{\partial p}{\partial x}) \quad \text{and} \quad \tilde{v} = \hat{v} + D (B_v - \frac{\partial p}{\partial y}) \tag{11}$$

where $D = V_P / A_P$. The pressure gradients in equations (11) are evaluated locally for each element. The discrete continuity equation is obtained from

$$\nabla \cdot \tilde{\mathbf{u}} = 0 \tag{3}$$

written as

$$\int_S (\tilde{\mathbf{u}} \cdot \mathbf{n}) \, dS = 0 \tag{12}$$

The values of D at points within the element are linearly interpolated from the nodal values. The pressure gradients $(\partial p/\partial x)$ and $(\partial p/\partial y)$ are now local at the cell faces, and can be related to the nodal pressures (p_P, p_A, p_B) because of the linear interpolation used. If the equations for $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$ are substituted in the two interface flux relations, the contributions from element PAB to the continuity at node P are obtained. Similar contributions from all elements surrounding P then provide a pressure equation at P given by

$$A^p_P \, p_P = \Sigma A^p_{nb} \, p_{nb} + M_P \qquad (13)$$

where M_P is the source term arising from the terms containing $\hat{u}$, $\hat{v}$ and B_u, B_v. We now seek a solution (u, v, p) that satisfies the set of discrete equations (6), (8) and (13).

3. SINGLE GRID SOLUTION STRATEGY AND PERFORMANCE

The system of coupled equations (6), (8) and (13) has been previously solved by a sequential solution method, SIMPLER [19]. The iterative update involves solving in a cycle the pressure equation, followed by the two momentum equations. Starting from guessed velocity and pressure fields, the coefficients A_P and A_{nb} are first assembled. Using these, the pressure equation is assembled through the above mentioned formulae. The pressure equation is then solved by any convenient linear solver. For simplicity, we have used a point Gauss-Seidel scheme, which is repeated a few (nswpp) times. This pressure field is then used to solve the velocity equations. The previously assembled A_P and A_{nb} are used, and a few (nswpm) sweeps of the Gauss-Seidel scheme are made. The new velocity field is then used for calculating the next iterate of the pressure field.

A point to mention is the under-relaxation used to hold the iterative process from becoming unstable. This is done by adding only a part of the change to the flow variables in an implicit manner by modifying the central coefficients and the source terms in the discrete equations. Figure 2 shows the behavior of the single grid scheme for flow in a driven square cavity, discretized on a triangular grid with increasing number of elements. As is evident, the convergence deteriorates with increasing number of nodes, which significantly increases the cost of performing systematic mesh refinement studies.

4. DETAILS OF THE PRESENT MULTIGRID IMPLEMENTATION

Mesh generation and refinement

In the present procedure, the coarsest mesh is first generated as for any single grid procedure, by the Delaunay triangulation method. Subsequent finer grids are then generated by successively dividing each element into four elements (Figure 3(a)). A prespecified number of nested grids are thereby obtained. Each coarse grid element shares three nodes with the daughter finer grid elements. This grid arrangement makes the intergrid transfers as well as the construction of coarse grid equations simpler than with the practice of using different meshes for each grid density [4,5,7]. However, it has the disadvantage that the coarsest grid may not be very smooth. Nevertheless, the boundary shape is still accurately captured because during refinements, the daughter nodes are moved to coincide with the boundary shape.

The coarse grid discrete equations

Successful multigrid procedures rely heavily on consistent practices for the construction of the coarse grid equations and for the restriction and prolongation operators. Consistent restriction of variables and residuals to the coarser grids is the most important aspect of multigrid procedures for a system of equations, especially the fluid flow equations. For nonlinear equations, the Full Approximation Scheme (FAS) is the most suitable scheme for deriving the coarse grid equations. This is an extension of the more straight-forward Correction Scheme (CS) that is used for linear equations.

Consider the discrete fine grid equations given by

$$\mathbf{L}^{f}\,\mathbf{q}^{f} = \mathbf{F}^{f} \tag{14}$$

where $\mathbf{L}^{f}$ is the nonlinear operator matrix made of the convection and diffusion terms, $\mathbf{q}^{f}$ is the solution vector, and $\mathbf{F}^{f}$ is the right-hand side vector. The superscript f is used to denote the fine grid. After a few iterations on the fine grid, the residual is computed as

$$\mathbf{R}^{f} = \mathbf{F}^{f} - \mathbf{L}^{f}\,\mathbf{q}^{f} \tag{15}$$

This residual is restricted to the next coarser grid, and it is required that the corrections satisfy the equation

$$\mathbf{L}^{f-1}\,\Delta\,\mathbf{q}^{f-1} = \mathrm{I}_{f}^{f-1}\,\mathbf{R}^{f} \tag{16}$$

where $\mathbf{L}^{f-1}$ is the nonlinear operator on the coarse grid, $\Delta\,\mathbf{q}^{f-1}$ is the vector of corrections on the coarse grid, and I_{f}^{f-1} is the restriction operator. For the FAS scheme, equation (16) is rewritten as

$$\begin{aligned}\mathbf{L}^{f-1}\,(\Delta\,\mathbf{q}^{f-1} + \mathrm{I}_{f}^{f-1}\,\mathbf{q}^{f}) &= \mathrm{I}_{f}^{f-1}\,\mathbf{R}^{f} + \mathbf{L}^{f-1}\,(\mathrm{I}_{f}^{f-1}\,\mathbf{q}^{f})\\ &= \mathbf{F}^{f-1} + \mathrm{I}_{f}^{f-1}\,\mathbf{R}^{f} - (\mathbf{F}^{f-1} - \mathbf{L}^{f-1}\,(\mathrm{I}_{f}^{f-1}\,\mathbf{q}^{f}))\end{aligned} \tag{17}$$

or

$$\mathbf{L}^{f-1}\,\mathbf{q}^{f-1} = \mathbf{F}^{f-1} + (\mathrm{I}_{f}^{f-1}\,\mathbf{R}^{f} - \mathbf{R}_{0}^{f-1}) \tag{18}$$

where $\mathbf{R}_{0}^{f-1}$ is the residual on the coarse grid, calculated using the restricted solution vector and $\mathbf{q}^{f-1}$ is the solution on the coarse grid. After a fixed number of iterations on the coarse grid, the corrections implied by the coarse grid solution can be extracted from the relation

$$\Delta\,\mathbf{q}^{f-1} = \mathbf{q}^{f-1} - \mathrm{I}_{f}^{f-1}\,\mathbf{q}^{f} \tag{19}$$

The above FAS scheme is used in a straight-forward way for the momentum equations. The restriction and prolongation operators defined below provide a consistent and convergent multigrid procedure. The main complexity in the present scheme lies in the construction of the pressure equation which satisfies mass continuity not for the nodal velocities but for a different set of fluxes implicitly located at the cell faces of the control volume. As the success of the present procedure relies solely on this aspect, we give below details of the coarse grid pressure equation.

The FAS form of the coarse grid pressure equation that results from the continuity satisfaction condition is derived as follows. We begin with the correction equation

$$(\nabla \cdot \tilde{u}')^{f-1} = I_f^{f-1} R_c^{\,f} \tag{20}$$

where the prime denotes the correction in $\tilde{u}$, and the right-hand side is the restricted residual in the continuity equation. Equation (20) is expressed as

$$\nabla \cdot (\tilde{u} + \tilde{u}')^{f-1} = I_f^{f-1} R_c^{\,f} + (\nabla \cdot \tilde{u})^{f-1} \tag{21}$$

Now,

$$\tilde{u} = \hat{u} + D\,\tilde{\nabla}p \quad \text{and} \quad \tilde{v} = \hat{v} + D\,\tilde{\nabla}p \tag{22}$$

where $\hat{u}$ is the momentum velocity and $\tilde{\nabla}p$ is the pressure gradient that is used to evaluate the cell face fluxes. For the coarse grid equations, the components of $\hat{u}$ are defined as

$$\hat{u} = (R_u + \Sigma A_{nb}\, u_{nb}) / A_P + (1 - \alpha)\, u$$

and

$$\hat{v} = (R_v + \Sigma A_{nb}\, v_{nb}) / A_P + (1 - \alpha)\, v \tag{23}$$

where R_u and R_v are the net coarse grid momentum residuals defined from equation (21) as

$$R = I_f^{f-1} R^f - R_0^{\,f-1} \tag{24}$$

Substituting equations (22) in (21), the coarse grid continuity equation is given by

$$\nabla \cdot (\hat{u} + D\,\tilde{\nabla}p + \hat{u}' + D\,\tilde{\nabla}p')^{f-1} = I_f^{f-1} R_c^{\,f} + \nabla \cdot (\hat{u} + D\,\tilde{\nabla}p)^{f-1} \tag{25}$$

where p^{f-1} is the restricted pressure $I_f^{f-1}\, p^f$. Equation (25) can be further rewritten as

$$\begin{aligned} \nabla \cdot (D\,\tilde{\nabla}p + D\,\tilde{\nabla}p')^{f-1} &= I_f^{f-1} R_c^{\,f} - \nabla \cdot \hat{u}^{f-1} + (\nabla \cdot D\,\tilde{\nabla}p + \nabla \cdot \hat{u})^{f-1} \\ &= I_f^{f-1} R_c^{\,f} - \nabla \cdot \hat{u}^{f-1} + R_{c0}^{\,f-1} \end{aligned} \tag{26}$$

where $R_{c0}^{\,f-1}$ is the coarse grid residual in the pressure equation calculated using the restricted values of the variables. It must be noted that because of the segregated method of solution, $\hat{u}'$ is set to zero for the pressure equation. Now, in the FAS practice, the left-hand side terms of equation (26) can be combined to give

$$\nabla \cdot (D\,\tilde{\nabla}p)^{f-1} = -\nabla \cdot \hat{u}^{f-1} + R_c^{\,f-1} \tag{27}$$

where p^{f-1} is now redefined to be

$$p^{f-1} = I_f^{f-1}\, p^f + (p')^{f-1} \quad \text{and} \quad R_c^{\,f-1} = I_f^{f-1} R_c^{\,f} + R_{c0}^{\,f-1} \tag{28}$$

Equation (27) has the standard structure of the pressure equation with an added residual R_c^{f-1}

Restriction and prolongation operations

Restriction and prolongation operators for structured rectangular and curvilinear grids are now well established. For arbitrarily generated sequence of unstructured grids the intergrid transfers must be performed through systematic interpolations using appropriate geometric coordinates of the variable locations [2]. An advantage of constructing fine grids embedded within the coarse grids is that the simple injection scheme can be used as the restriction operator for the nodal variables. Thus coarse grid values for (u, v, p) are obtained by locating the fine grid daughter nodes coincident with the considered coarse grid nodes.

For the residuals in the momentum equations, several fine grid residuals are summed to obtain the corresponding coarse grid residual $I_f^{f-1} \mathbf{R}^f$. We need to determine the fractions of the fine grid control volumes around a coarse grid node that contribute to the coarse grid control volume (see Figure 3(b)). The coarse grid control volume around P in two dimensions is given by the area ABCDEFGHIJKL. This is composed of fractions of the fine grid control volumes around each of the nodes P, A, B ... and L. It is apparent that the complete fine grid control volume around P contributes to the coarse grid volume. It can be shown that the rest of the coarse grid volume is made of the sum of half the fine grid volumes around each of the nodes A, B, ...and K. Therefore, the restricted residual at point P is the sum of the fine grid residual at point P, and half the fine grid residuals at the surrounding fine grid nodes.

The prolongation process similarly is considerably simplified because of the mesh embedding. Coarse grid corrections to the solution are prolongated by direct injection at those fine grid nodes that coincide with the coarse nodes. For those fine grid nodes that lie in between the coarse nodes, the corrections are determined as averages of the corrections at the two surrounding coarse nodes. For example, in Figure 3(a), the coarse grid corrections at nodes P, A, and B are injected onto the next finer grid, whereas the corrections at a node such as D are determined as averages of the corrections at P and A.

5. TEST CALCULATIONS

We shall now present the performance of the algorithm in three model flow problems that illustrate the potential of the technique in calculating complex internal flows. The three selected problems reflect complex geometry, elliptic nature of the flow field and the presence of very fine scale variations in the flow that can only be resolved by a very fine mesh. In future, other problems that contain inflows and outflows, periodic boundary conditions and turbulence equations will be considered. The main point to be demonstrated here is that the method converges rapidly and that the rate of convergence is independent of the mesh density. In comparison with the single grid convergence shown in Figure 2, the multigrid method should save a large number of iterations. This is indeed the case as will be presented below.

Laminar Flow in a Square Cavity

We have conducted a systematic testing of the influence of the flow Reynolds number, the under-relaxation factors and the mesh density for three model driven cavity problems. The first one is the familiar problem of flow in a driven square cavity. In our tests, the square cavity is discretized by triangular elements. The triangulation is performed by the Delaunay procedure. Several levels of grid are then superimposed over the coarsest grid. Since upwinding was not used in the present study, for each mesh level, there was a limiting value of the Reynolds number beyond which convergence was not possible. Therefore, in the multigrid sequence, the desired Reynolds number was used only on the finest mesh. Iterations on each of the coarser meshes were performed with its stable maximum value of the Reynolds number, following along the concept of double discretisation. Two fixed V- cycles were examined. In the first, the number of iterations on the coarse grids increased as the coarsest grid was approached. On the locally finest grid, one iteration was performed. The next grid used two relaxations and the subsequent one three and so on. The same number of relaxations were performed on the up-leg of the V-cycle, except at the top of the V-cycle. In the second fixed cycle, a fixed number of three coarse grid relaxations were performed accompanied by one relaxation on the finest grid. Both schemes were well convergent except for minor differences in the rates of convergence and the CPU times.

Figure 4 shows the convergence history for a Reynolds number of 50 for different mesh densities, with the mass residual plotted against the number of iterations on the finest grid. In all the runs, the coarsest grid had 40 elements and 29 nodes. The finest grid in the 5-grid run had 10240 elements and 5249 nodes. It is apparent from the plots that the rate of convergence in all cases is nearly independent of the grid size. There is a five order decrease in the mass residual in less than 20 multigrid cycles. This may be compared with the convergence shown (for 640 elements) if only a single grid is used. Figure 5 shows the multigrid convergence for the highest permitted Reynolds number of 500 which requires a slightly larger number of iterations due to the increased nonlinearity. The calculated results agreed well with previously reported results of Ghia et al. [20] and Vanka [13].

Laminar flow in a triangular cavity

The flow in a triangular cavity wherein the fluid motion is set by the motion of the top wall is an interesting complex flow which results in an infinite number of vortices of diminishing intensity towards the lower corner of the cavity [21, 22]. Although the square cavity has been studied extensively, there has been very little numerical work reported on the triangular cavity [23]. The triangular cavity cannot be easily discretized by a curvilinear mesh that is smooth and has high quality. However, it is ideally suited for triangulation. For the calculations presented here, the depth of the cavity is twice the width of the top wall. Here, as in the square cavity, the top wall is moved to the right with a velocity u = 1. A series of Reynolds numbers up to 800 were considered and the performance of the method was evaluated. Here the Reynolds number is defined with respect to the depth of the cavity and the top wall velocity.

Figures 6 and 7 show the multigrid convergence of the code for Reynolds numbers of 50 and 800. Linear convergence is observed even with 12288 elements and 6305 nodes. The velocity vectors and streamtraces in the flow field are shown in Figures 8 and 9 for Reynolds numbers of 50 and 800. The occurrence of the series of vortices is replicated by the calculations to the point of grid resolution. Further resolution near the bottom corner should reveal more and more eddies of smaller dimension. Moffat [21] has shown that for Stokes flow, the distance of each eddy from the corner increases in geometric progression as does its intensity. This was indeed seen for all the eddies except for the one near the top wall. Therefore, starting from the second eddy, the ratios of successive distances from the corner for Re = 50 are respectively, 1.97, 1.98 and 1.9. The deviation from the expected series for the topmost eddy is probably because of the breakdown of the Stokes flow assumption there. Near the top wall, inertial effects dominate, and Moffat's analysis is not valid there.

Laminar flow in a semicircular cavity

The final problem considered is the flow in a semi-circular cavity which has a curved boundary. In this case, the coarsest triangulation does not capture the true shape of the boundary. However, as the mesh is refined, the fine grid points are moved to the boundary to fit the shape. Thus a better representation of the boundary is obtained. For this geometry also, several Reynolds numbers and mesh densities were considered. As a representative plot, Figure 10 shows the convergence for the Reynolds number of 500 discretized with 3584 elements and 1873 nodes. The consistency of the coarse grid and fine grid transfers is demonstrated by this rate of convergence. It is to be noted that only the near boundary elements are altered and no remeshing is performed. This preserves the restriction/prolongation practices that are valid in the interior. The velocity vectors and streamtraces in the flow field for Re = 500 are shown in Figure 11.

Table 1 summarizes all the calculations currently performed with this procedure. The corresponding work units are,also presented, which accounts for the coarse grid iterations. The work involved in the injections and interpolations during restriction and prolongation is neglected as per the standard practice in multigrid literature.

6. CONCLUSIONS

In this paper, a multigrid method for unstructured grids based on geometric coarsening (versus algebraic coarsening, Webster [14]) has been presented. A sequence of embedded grids has been used to smooth out low frequency errors, and accelerate the convergence on fine grids. The momentum and continuity equations are discretized by a control volume procedure with equal order interpolations for the variables. The mass continuity equation is transformed to a pressure equation which is derived through special interpolations that provide a well-connected pressure field. A simple iterative scheme such as the Gauss-Seidel method has been used to relax the discrete equations on any grid. The coarse grid pressure equation is constructed by a consistent restriction of the cell face fluxes and appropriate equations. It is demonstrated that the method provides good multigrid convergence in the three test problems for all Reynolds numbers up to the value permitted by the cell Reynolds number criterion of the central differencing scheme. Future extensions to this procedure are underway to include periodic boundary conditions, turbulence models, time-dependent terms, and three-dimensional variations.

ACKNOWLEDGEMENTS

We wish to thank Mr. Kevin Stone for his assistance in performing some of the calculations and in making the relevant plots.

REFERENCES

1. Brandt, A. (1977) Multi-Level Adaptive Solutions to Boundary-Value Problems, *Math. Comp.* **31**, No. 138, 333-390.
2. Löhner, R. and Morgan, K. (1987) An Unstructured Multigrid Method for Elliptic Problems, *Int. J. Num. Meth. Eng.* **24**, 101-115.
3. Koobus, B., Lallemand, M. and Dervieux, A. (1994) Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation, *Int. J. Num. Meth. Fluids* **18**, 27-42.
4. Mavriplis, D. J. and Jameson, A. (1987) Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes, *ICASE Rep. No.* **87-53**, NASA CR-178346.
5. Mavriplis, D. J. (1988) Multigrid Solution of the 2-D Euler Equations on Unstructured Triangular Meshes, *AIAA J.* **26**, No. 7, 824-831.
6. Lallemand, M., Steve, H. and Dervieux, A. (1992) Unstructured Multigridding by Volume Agglomeration: Current Status, *Comput. Fluids* **21**, No. 3, 397-433.
7. Parthasarathy, V. and Kallinderis, Y. (1994) A New Multigrid Approach for 3D Unstructured, Adaptive Grids, *AIAA*-94-0078.
8. Mavriplis, D. J. and Jameson, A. (1990) Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes, *AIAA J.* **28**, No. 8, 1415-1425.
9. Baliga, B. R. and Patankar, S. V. (1983) A Control Volume Finite-Element Method for Two-Dimensional Fluid Flow and Heat Transfer, *Num. Heat Transfer.* **6**, 245-262.
10. Prakash, C and Patankar, S. V. (1985) A Control Volume-Based Finite-Element Method for Solving the Navier-Stokes Equations Using Equal-Order Velocity-Pressure Interpolation, *Num. Heat Transfer* **8**, 259-280.

11. Prakash, C. (1986) An Improved Control Volume Finite-Element Method for Heat and Mass Transfer, and for Fluid Flow Using Equal-Order Velocity-Pressure Interpolation, *Num. Heat Transfer* **9**, 253-276.
12. Hookey, N. A., Baliga, B. R. and Prakash, C. (1988) Evaluation and Enhancements of Some Control Volume Finite Element Methods - Part 2. Incompressible Fluid Flow Problems, *Num. Heat Transfer* **14**, 273-293.
13. Vanka, S. P. (1986) Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables, *J. Comput. Physics* **65**, No. 1, 138-158.
14. Webster, R.(1994) An Algebraic Multigrid Solver for Navier-Stokes Problems, *Int. J. Num. Meth. Fluids* **18**, 761-780.
15. Weatherill, N. P. (1988) A Method for Generating Irregular Computational Grids in Multiply Connected Planar Domains, *Int. J. Num. Meth. Fluids* **8**, 181-197.
16. Smith, K. M., Cope, W. K. and Vanka, S. P. (1993) A Multigrid Procedure for Three-Dimensional Flows on Non-Orthogonal Collocated Grids, *Int. J. Num. Meth. Fluids* **17**, 887-904.
17. Rhie, C. M. and W. L. Chow (1983) Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, **21**, No. 11, 1525-1532.
18. Peric, M., Kessler, R. and Scheurer (1988) Comparison of Finite Volume Numerical Methods with Staggered and Collocated Grids, *Comput Fluids* **16**, 389-903.
19. Patankar, S. V. (1980) *Numerical Heat Transfer.* Hemisphere, Washington., D. C.
20. Ghia, U., Ghia, K. N. and Shin, C. T. (1982) High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. Comput. Physics* **48**, No. 3, 387-411.
21. Moffat, H. K. (1963) Viscous and Resistive Eddies Near a Sharp Corner, *J. Fluid Mech.* **18**, 1-18.
22. Batchelor, G. K. (1956) On Steady Laminar Flow with Closed Streamlines at Large Reynolds Number, *J. Fluid Mech.* **1**, 177-190.
23. Ribbens, C. J., Watson, L. T. and Wang, C. Y. (1994) Steady Viscous Flow in a Triangular Cavity, *J. Comput. Physics* **112**, No. 1, 173-181.

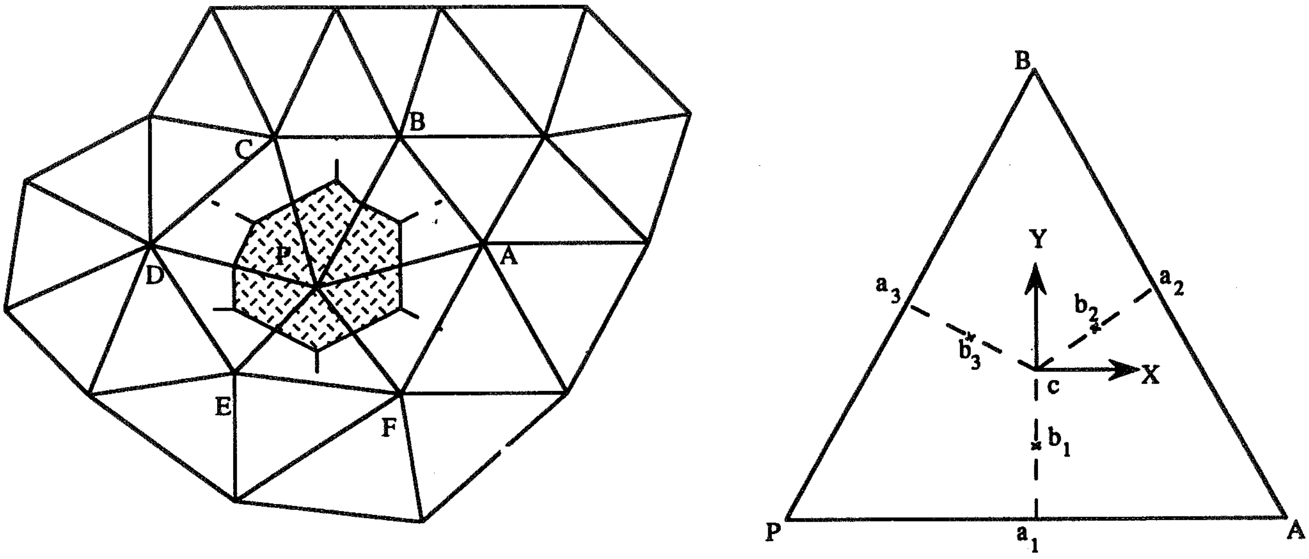


Figure 1: (a) Unstructured mesh with control volume around node P; (b) Element PAB and local coordinate system

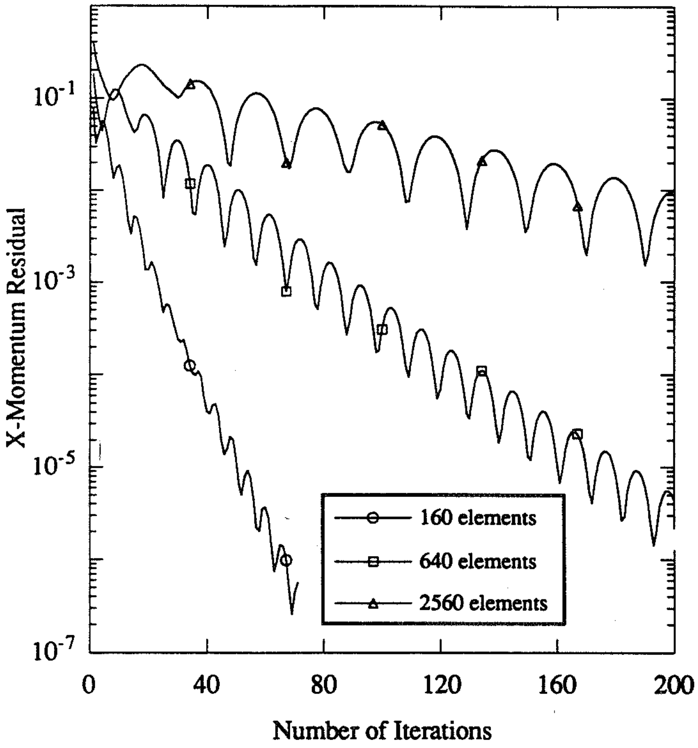


Figure 2: Single grid convergence for shear driven flow in a square cavity with increase in number of elements

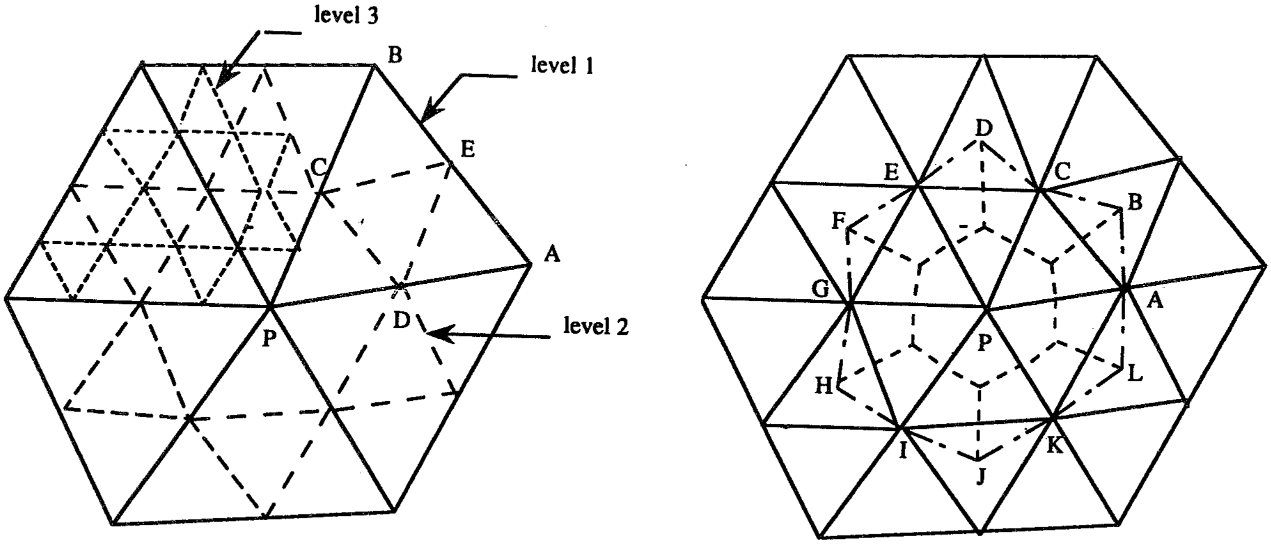


Figure 3: (a) Mesh refinement; (b) Course and fine grid control volumes around node P

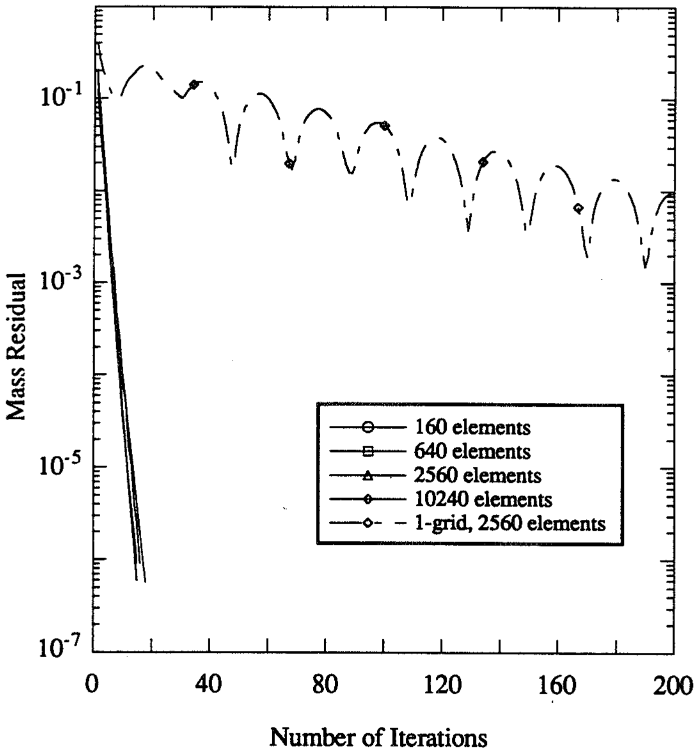


Figure 4: Muitigrid and single grid convergence for laminar flow in a square cavity at Re = 50

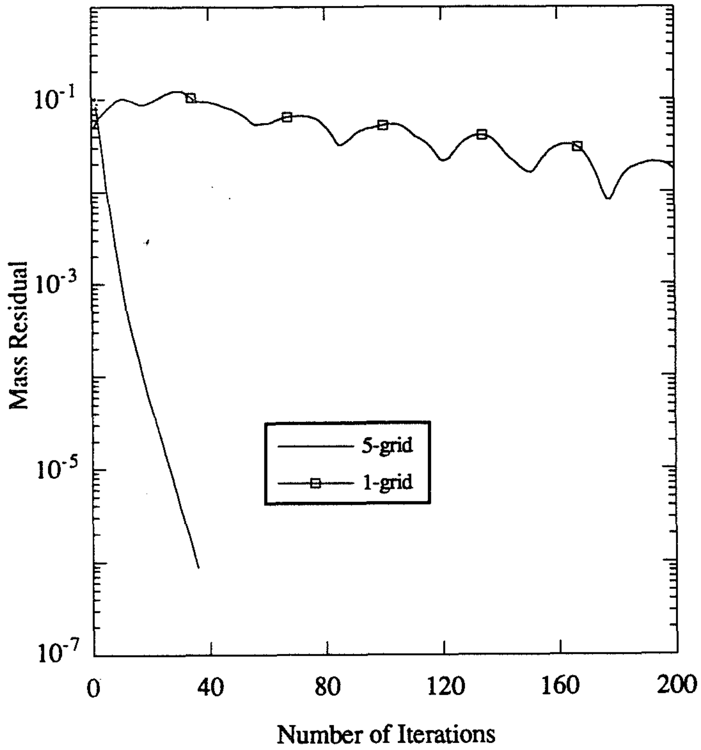


Figure 5: Multigrid and single grid convergence for laminar flow in a square cavity at Re = 500

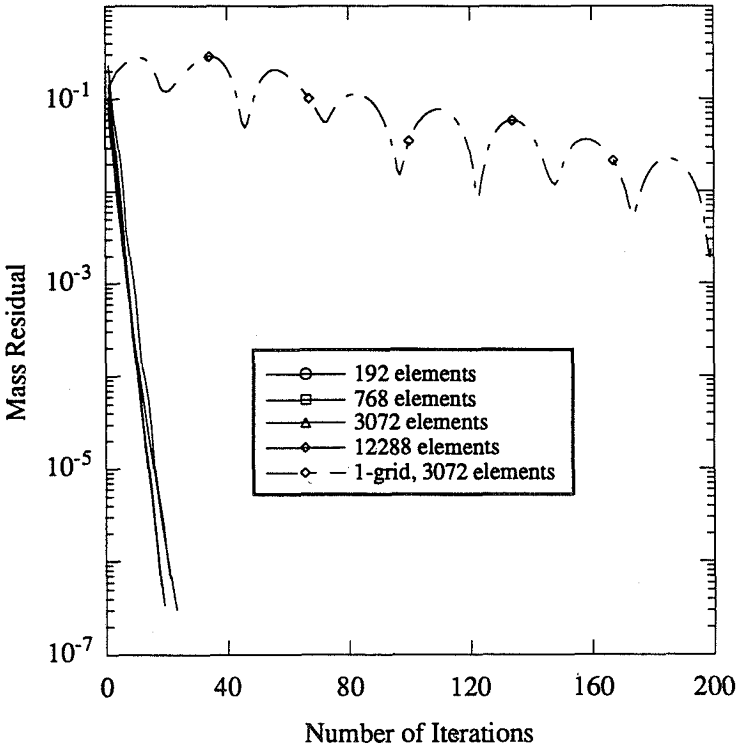


Figure 6: Multigrid and single grid convergence for laminar flow in a triangular cavity at Re = 50

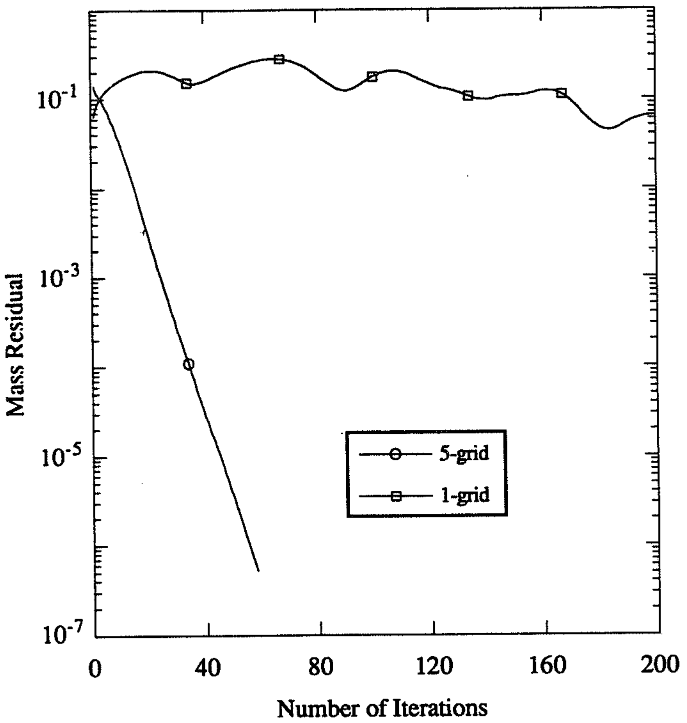


Figure 7: Multigrid and single grid convergence for laminar flow in a triangular cavity at Re = 800, with 12288 elements

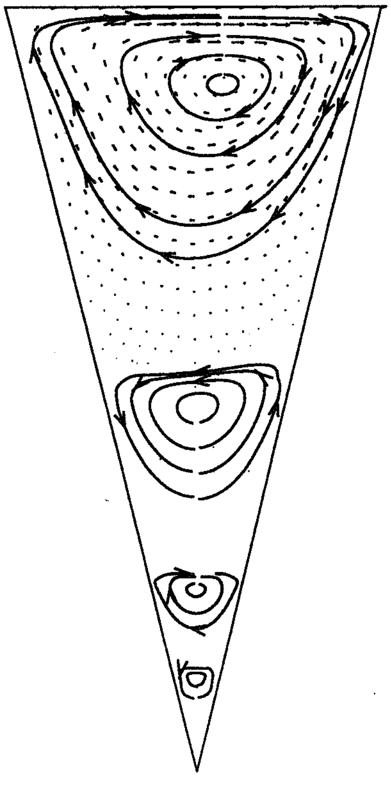


Figure 8: Velocity vectors and streamtraces for laminar flow in a triangular cavity at Re = 50

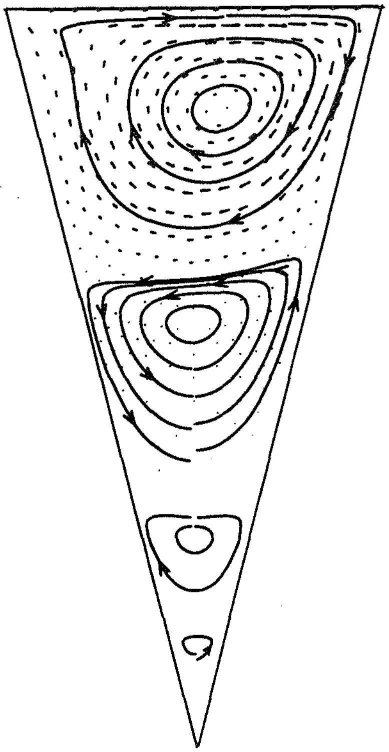


Figure 9: Velocity vectors and streamtraces for laminar flow in a triangular cavity at Re = 800

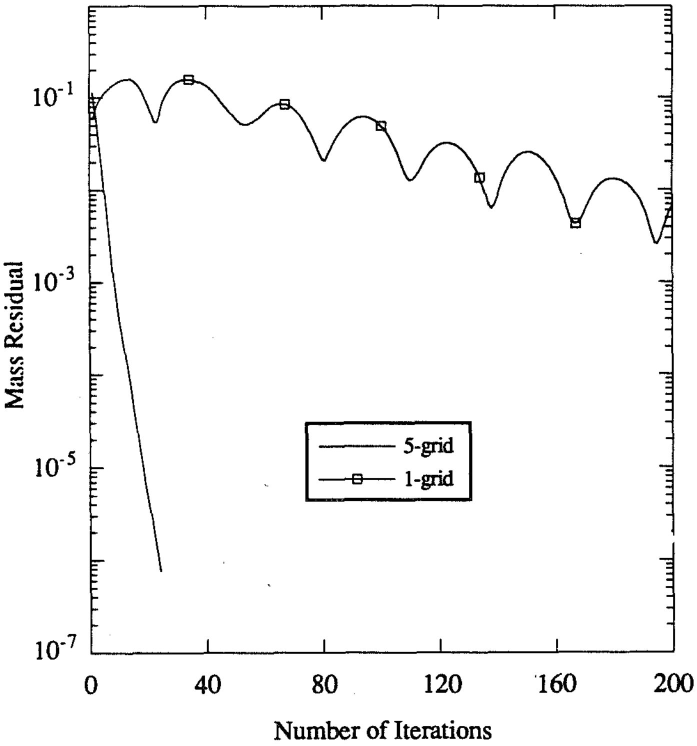


Figure 10: Multigrid and single grid convergence for laminar flow in a semicircular cavity at Re = 500, with 3584 elements

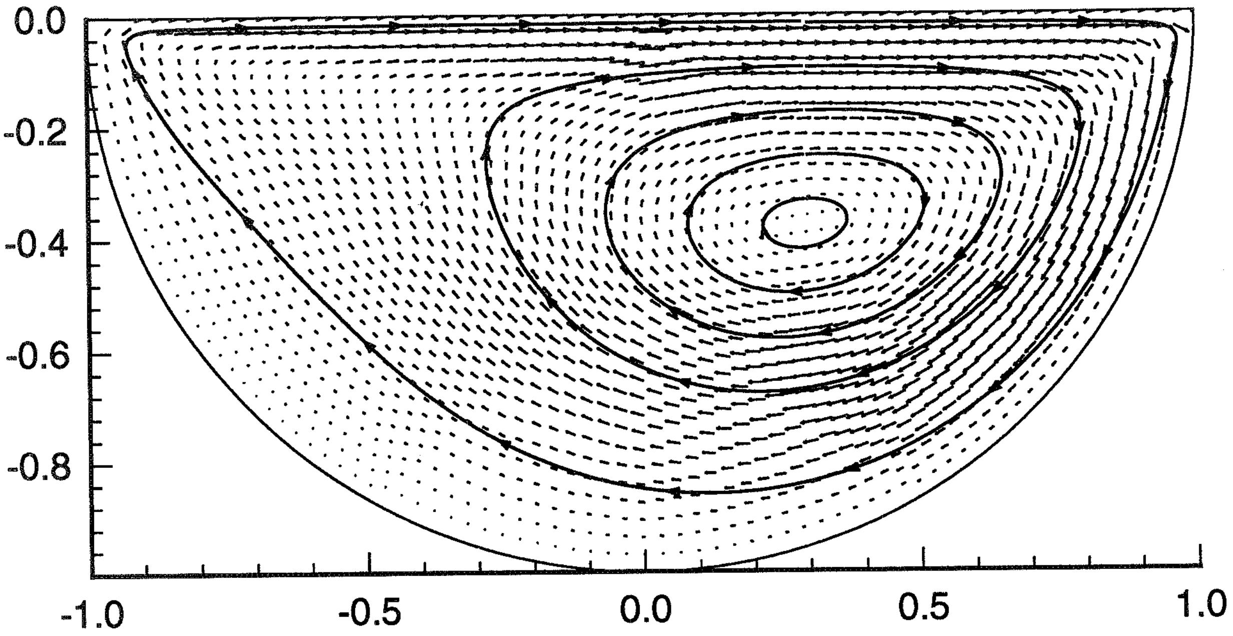


Figure 11: Velocity vectors and streamtraces for laminar flow in a semicircular cavity at Re = 500

Table 1: Number of fine grid iterations for a five order decrease in the residuals, shown as a function of the number of elements and the Reynolds number. Each fine grid iteration corresponds to three work units

Reynolds number Elements	50	100	200	500	600
Square Cavity					
160	16	22	-	-	-
640	15	20	30	-	-
2560	15	19	25	-	-
10240	18	17	29	36	51
Triangular Cavity					
192	21	21	-	-	-
768	18	17	26	-	-
3072	19	16	24	-	-
12288	23	17	16	37	50
Semicircular cavity					
56	14	18	-	-	-
224	1	16	25	-	-
896	14	18	24	-	-
3584	14	15	23	24	45

The Multigrid-Mask Numerical Method for Solution of Incompressible Navier-Stokes Equations

Hwar-Ching Ku

Johns Hopkins University Applied Physics Laboratory

Johns Hopkins Road, Laurel, MD 20723

Aleksander S. Popel

Johns Hopkins University School of Medicine

Department of Biomedical Engineering

720 Rutland Avenue, Baltimore, MD 21205

Abstract

A multigrid-mask method for solution of incompressible Navier-Stokes equations in primitive variable form has been developed. The main objective is to apply this method in conjunction with the pseudospectral element method solving flow past multiple objects. There are two key steps involved in calculating flow past multiple objects. The first step utilizes only Cartesian grid points. This homogeneous or mask method step permits flow into the interior rectangular elements contained in objects, but with the restriction that the velocity for those Cartesian elements within and on the surface of an object should be small or zero. This step easily produces an approximate flow field on Cartesian grid points covering the entire flow field. The second or heterogeneous step corrects the approximate flow field to account for the actual shape of the objects by solving the flow field based on the local coordinates surrounding each object and adapted to it. The noise occurring in data communication between the global (low frequency) coordinates and the local (high frequency) coordinates is eliminated by the multigrid method when the Schwarz Alternating Procedure (SAP) is implemented.

Two dimensional flow past circular and elliptic cylinders will be presented to demonstrate the versatility of the proposed method. An interesting phenomenon is found that when the second elliptic cylinder is placed in the wake of the first elliptic cylinder a traction force results in a negative drag coefficient.

1 Introduction

The motive to develop the multigrid-mask method is to remedy the drawback of grid generation which often results in a tremendous effort to achieve the desired layout of grid points for flow past multiple objects. As expected, the grid generation becomes even more difficult when the objects are close to each other or randomly moving. The situation occurs in many physical problems, such as cross flow in shell-tube heat exchangers, two phase flow in multiple particle sedimentation, and flow of blood cells in arteriols, capillaries, and venules (Stokes flow).

The conventional numerical simulation of Navier-Stokes (or Stokes) flow with multiobject systems falls into two main categories: (I) distinguishable and (II) indistinguishable fluid-object interfaces. Category I defines a distinct boundary between objects and fluid, and exact boundary conditions; velocity and force can be prescribed on the surface of objects. Actually, this category partitions the entire flow domain into two heterogeneous systems: objects (may or may not have fluid inside) and fluid system. It is capable of providing highly accurate details of flow interaction among objects but is computationally intensive (not more than three objects). Ingber [1] and Tran-Cong & Phan-Thien [2] use the boundary element method for suspensions of rigid particles in Stokes flow and Li, Zhou, & Pozrikidis [3] use the boundary element method for deformable particles.

Category II implies that a fuzzy boundary exists between objects and fluid. In other words, there is no distinct boundary between objects and fluid; therefore, a homogeneous system can be applied to the entire domain. As a result, a single set of fluid dynamics equations holds at all grid points (a "stationary" grid) of the domain and no internal boundaries are necessarily defined, i.e., original boundary conditions, force on the fluid-object surfaces, now become the additional inhomogeneous source term in the Navier-Stokes equations. However, a sharp discontinuity for the velocity field (or other variables) between the fluid-object interfaces should be preserved in conformity with the original problem. In order to maintain a sharp front between fluid-object interfaces, the fuzzy boundary should be restricted to within a few mesh distances; the less the mesh distance, the better the resolution of fluid-object interfaces. A variety of means to achieve the desired sharp fluid-object interface are suggested by many investigators [4, 5, 6]. Basically, the flow field is discretized by the finite difference approximation on a stationary grid to cover the entire flow domain. For the moving or deformable objects, a separate object grid which configures the geometry of objects needs to be defined, and this object grid is allowed to move with the speed interpolated from the stationary grid. The discussion of moving or deformable cases is beyond the current scope.

Briscolini and Santangelo [5] proposed the spectral method to solve the incompressible unsteady flow over a circular cylinder by introducing a strip zone (or equivalent to stationary boundary layers in which a steep change of field variables occurs) of control within a few meshes. A narrow mask (Gaussian) function, defined as zero inside the objects and one elsewhere along with a smooth connection between these two values within the strip zone, is applied to the velocity field. The drawback of the mask method is that it only provides an approximate flow field due to an inexact capturing of the configuration of the objects by a stationary grid alone as well as the thickness of the fuzzy boundary (a few meshes wide) between the fluid and cylinder. Peskin [6] adopted the immersed boundary method for numerical simulation of blood flow in the human heart. His idea is very similar to the mask method of Briscolini and Santangelo [5] except a separate material grid is added to trace the heart wall movement. For the data communication between the stationary grid and the material grid, Peskin [6] employs an approximation to the delta-function to define the interpolated velocity and force transferred between the fluid-object system.

The objective of this paper is to develop a numerical method which combines the desired features of both category I and II and that can also accurately simulate the flow interaction among multiple objects. In practice, it includes two major steps: (1) apply a stationary grid to obtain a fast solution covering the entire domain, which is similar to the category II approach but differs in some respects by requiring that the velocity for the stationary grid falling inside objects is imposed to be small or zero (a homogeneous step or mask method is hereafter named); and (2) generate a local fluid grid surrounding objects to exactly capture the surface configuration of objects, which is similar to category I by prescribing exact boundary conditions on the surface of objects (a heterogeneous step). Notice that step (1) only provides an approximate flow field and step (2) corrects the approximate flow field predicted from step (1) with the imposition of exact boundary conditions on the surface of objects.

In domain overlapping terminology, one can regard the local fluid (or fine) grid as being fully overlapped with the global stationary (or coarse) grid. A data communication process between the stationary and fluid grid can be conducted by the Schwarz Alternating Procedure (SAP) [7]. Although the grid points of each grid system in the overlapping area are not coincided with each other, the SAP iterative scheme still can be used effectively for data communication between the stationary and fluid grid in conjunction with the multigrid method [8, 9]. The role of the multigrid method in the SAP process ensures a smooth data interpolation between the global stationary and local fluid grid without introducing any high-frequency error.

The solution of the Navier-Stpkes equations is implemented by the pseudospectral element method, which is an extension of the global pseudospectral method to the element-type method by requiring that the function continuity c^0 be continuous across the interface between two adjacent elements [10] when calculating the derivatives of a function.

2 Primitive Variable Formulation

2.1 Navier-Stokes Equations

In tensor notation, the time-dependent Navier-Stokes equations in dimensionless form can be described as

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_j^2} \tag{1a}$$

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{1b}$$

Here u_i is the velocity component and Re is the Reynolds number.

The method applied to solve the Navier-Stokes equations is Chorin's [11] splitting technique. According to this technique, the equations of motion are written in the form

$$\frac{\partial u_i}{\partial t} + \frac{\partial p}{\partial x_i} = F_i \tag{2}$$

where $F_i = -u_j \, \partial u_i / \partial x_j + 1/\mathrm{Re} \, \partial^2 u_i / \partial x_j^2$.

The first step is to split the velocity into a sum of predicted and corrected values. The predicted velocity is determined by time integration of the momentum equations without the pressure term

$$\bar{u}_i^{n+1} = u_i^n + \Delta t F_i^n. \tag{3}$$

The second step is to determine the pressure and corrected velocity fields that satisfy the continuity equation by using the relationships

$$u_i^{n+1} = \bar{u}_i^{n+1} - \Delta t \frac{\partial p}{\partial x_i} \tag{4a}$$

$$\frac{\partial u_i^{n+1}}{\partial x_i} = 0. \tag{4b}$$

Here the superscript n denotes the n-th time step.

An equation for the pressure can be obtained by taking the divergence of Eq. (4a). In view of Eq. (4b), we obtain

$$\frac{\partial^2 p}{\partial x_i^2} = \frac{1}{\Delta t} \frac{\partial \bar{u}_i}{\partial x_i}. \tag{5}$$

Note that the pressure solution on the global stationary grid is solved numerically by separation of variables [7], while the Generalized Conjugate Residual (GCR) method [12] is used to iteratively solve the pressure equation on the local fluid grid.

3 Domain Decomposition with Multigrid-Mask Method

As mentioned in the section of Introduction, two major steps are involved for the calculation of flow past multiple objects: a homogeneous step as well as a heterogeneous step. Data communication between the stationary and fluids grid by the multigrid method will be described in the process of the heterogeneous step. Each step is addressed as follows.

3.1 Mask Method - Homogeneous Step

A single coordinate system is used to produce a stationary grid to cover the entire fluid-object domain. Usually, several stationary grid points are contained inside the objects. This homogeneous step is sometimes called the mask method, which is analogous to that proposed by Briscolini & Santangelo [5] and Peskin [6]. In other words, it permits flow into the interior stationary grid points contained in the objects and considers the objects as a homogeneous (whole) system; no distinction between the fluid and objects is made. But the requirement that the velocity on the stationary grid points confined in the objects being small or zero should be met.

According to this step, the Cartesian grid points can be extended to cover the interior of each object and the entire domain. Such an approach enables us to take advantage of the fast solution for the operator resulting from the desired feature of a complete Laplacian type.

As pointed out in the Introduction, the mask method only provides an approximate flow field because the Cartesian grids contained in the objects cannot accurately represent the configuration of objects themselves. Besides, the flow field on the Cartesian grid points inside or on the surface of the objects should be prescribed in order to comply with the original problem, i.e., no flow or small velocity inside the objects (including on the surface).

Such a criterion, equivalent to finding a predicted velocity $\bar{u}^{n+1}$ inside the objects as appeared in Eq. (4a), can be met by setting

$$\bar{u}_i^{n+1} = u_i^p + \Delta t \frac{\partial p}{\partial x_i} \tag{6}$$

on the Cartesian grid points confined in the interior of objects. Here superscript p refers to the prescribed velocity. Presumably, this should implicitly force u^{n+1} to be equal to the prescribed value. However, due to the nonsmooth flow field exhibited around the fluid-object interfaces, simply choosing the predicted velocity $\bar{u}^{n+1}$ to be zero or constant does not guarantee that the velocity u^{n+1} obtained from Eq. (4a) be u_i^p inside the objects after solving Eq. (5). Thus, the predicted velocity $\bar{u}^{n+1}$ inside the objects can be obtained by the repeated solution of Eqs. (5) and (6). Usually, only 1 to 2 iterations are required to ensure that $\| u^{n+1} - u^p \| < 10^{-4}$ after a few hundred time steps.

3.2 Multigrid Method - Heterogeneous Step

In order to correct the approximate flow field predicted from the homogeneous step (based on the stationary grid), the heterogeneous step next accounts for the actual shape of the objects by adding their own local coordinates; an external fluid grid surrounds each object. Since the mask method does not define a distinct interface between fluid and objects, rather the fuzzy interface falls within a few meshes. As a result, such fluid-object interfaces need to be defined, and this is what the heterogeneous step tends to accomplish. The boundary conditions on the surface of objects are straightforward with no slip velocity.

In view of the domain decomposition approach for flow past multiple objects, one can regard the local subdomains (fine grid referred to the fluid grid surrounding each object) fully overlapped with the global (coarse grid referred to the stationary grid) rectangular domain as depicted in Fig. 1. As for the data communication between the fluid and stationary grid, the iterative SAP technique will be naturally suitable for this purpose, i.e., the global stationary grid provides the outer boundary information for the local fluid grid and in turn the local fluid grid corrects the flow field outside objects by imposing exact boundary conditions on the fluid-object interfaces.

Due to the different orientation and resolution of each grid system, simply exchanging the data through interpolation in the overlapping area, stationary (coarse)-fluid (fine) grid system, causes the high frequency error induced by the fine-grid (fluid grid) subdomain and hence affects the results throughout the whole computational domain. The technique of filtering the high-frequency noise is also known as the multigrid method. The coarse-grid correction process often used in the multigrid method is adopted in the overlapping area for the coupled pressure and velocity field and has been proposed by Ku & Ramaswamy [9]:

$$\nabla_c \cdot \mathbf{u}_c - \nabla_c \cdot (I_c^f \mathbf{u}_f) = I_c^f (r_f - \nabla_f \cdot \mathbf{u}_f). \tag{7}$$

Here $\nabla_c \cdot$ represents the operator of divergence on the coarse-grid subdomain, I_c^f is an interpolation operator from the fine-grid subdomain "f" to the coarse-grid domain "c," $\mathbf{u}$ is the velocity, and r_f is the divergence of the velocity field which should be set to zero at the first SAP iteration. The left hand side of Eq. (7) is the difference between the coarse-grid operator acting on the coarse-grid domain and the coarse-grid operator acting on the interpolated fine-grid subdomain (which is held fixed). When the term $\nabla_c \cdot \mathbf{u}_c$ appearing in Eq. (7) is substituted by Eq. (4a) the pressure equation in the coarse-grid domain is thus governed, and so is the pressure equation in the fine-grid domain. Actually, Eq. (7) implicitly functions as a coupled equation between the pressure and velocity; not only the residual of the right hand side of Eq. (7) should be equal to zero but also the unchanged velocity field during the SAP iteration is required.

In the overlapping area r_f cannot be predetermined and needs to be adjusted until the velocity field generated from the coupled pressure equations $\nabla_c \cdot \mathbf{u}_c = \nabla_c \cdot (I_c^f \mathbf{u}_f)$ and $\nabla_f \cdot \mathbf{u}_f = r_f$ is unchanged.

Once the residual $r_f - \nabla \cdot \mathbf{u}_f$ and velocity field do not change in the fine-grid subdomain, this implies that

$$\mathbf{u}_c = I_c^f \mathbf{u}_f. \tag{8}$$

Whenever either the residual $r_f - \nabla \cdot \mathbf{u}_f$ or the velocity field in the fine-grid subdomain still varies, Eq. (7) acts as a coarse-grid correction process to transfer the correction of the velocity field back to the fine-grid subdomain, i.e.,

$$\mathbf{u}_f^{new} = \mathbf{u}_f^{old} + I_f^c(\mathbf{u}_c - I_c^f \mathbf{u}_f^{old}). \tag{9}$$

This is vital for the success of the scheme. Changes in the velocity field are transferred back to the fine-grid subdomain rather than the velocity field itself. At each SAP iteration, r_f can be simply chosen as $r_f = \nabla_f \cdot \mathbf{u}_f^{new}$ from Eq. (7).

The multigrid-mask SAP iterative solution of the incompressible Navier-Stokes equations in primitive variable form for flow past multiple objects (also shown in Fig. 1) is summarized by the following algorithm:

1. First assume $\mathbf{u}^{n+1}$ on the outer boundary of each object. Usually $\mathbf{u}^n$ will be a good initial guess.

2. Solve the fine-grid or fluid grid system, where the pressure solution is obtained by the preconditioned General Conjugate Residual (GCR) method.

3. With the interpolated solution of $\mathbf{u}_c^{n+1}$ from step (2) through Eq. (8) in the overlapping area, solve the pressure on the coarse-grid domain (stationary grid) by the mask method with the eigenfunction expansion technique and also update $\mathbf{u}_f^{n+1}$ in the overlapping area of the fine-grid domain by the coarse-grid correction process in Eq. (9).

4. Repeat steps (2) & (3) until the velocity $\mathbf{u}^{n+1}$ in the overlapping area satisfies the convergence criterion of Eq. (8).

It is worthwhile to emphasize that even with strong discontinuity exhibited for the velocity on the grid points immediately outside the objects the multigrid-mask method indeed meets the requirements of both having small velocity inside the objects and satisfying of Eq. (8).

4 Results and Discussion

Four SAP iterations are employed for all the test problems, and the convergence criterion of Eq. (8) is satisfied by the requirement $\| \mathbf{u}_c - I_c^f \mathbf{u}_f \| \leq 2.5 \times 10^{-4}$. The radiation boundary condition [8] is applied on the truncated downstream to give the least influence upon the upstream flow development.

4.1 Circular cylinders

For the first benchmark test, we choose a uniform flow over a cylinder to give a comparison of results between the multigrid scheme and the pseudospectral element method [9], in which the computational domain is decomposed into two subdomains: an "O" grid domain, partially overlapped with the Cartesian grid domain. The diameter of a cylinder over the width of a channel is 1/20 in this numerical experiment; 18 × 15 elements (each element contains 7 × 7 points in the x and y directions) are allocated in the stationary grid system, and 15 × 6 elements in the fluid (or "O") grid system. The periodic character of the flow motion can be defined by the Strouhal number $S = fD/U_{max}$, where f is the shedding frequency, D is the diameter of a cylinder, and U is the maximum inlet velocity. Numerical results of drag coefficient C_D and lift coefficient C_L predicted by the multigrid-mask method, $1.379 \leq C_D \leq 1.394$, $-0.263 \leq C_L \leq 0.263$ for Re = 100 and $1.328 \leq C_D \leq 1.481, -0.733 \leq C_L \leq 0.733$ for Re = 250, are in good agreement with those calculated by the multigrid method of [9]: $1.36 \leq C_D \leq 1.385, -0.269 \leq C_L \leq 0.269$ for Re = 100 and $1.29 \leq C_D \leq 1.432, -0.711 \leq C_L \leq 0.711$ for Re = 250. The Strouhal number, $S = 0.168$ at Re = 100 and $S = 0.208$ at Re = 250, also reproduces the same results as those found in [9]. Streamline plots presented in Fig. 2 describe the typical flow motion behind the cylinder at Re = 100 and 250, respectively.

We secondly examine Poiseuille flow past multiple cylinders at Re = 20 using the multigrid-mask method. Figs. 3 and 4 show both the element layouts of the stationary and fluid grids and streamline plots for flow over four cylinders with the shortest distance 1.828 (Fig. 3a) and 0.414 (Fig. 4a) diameter of the cylinder. Numerical results indicate that less flow rate goes through the intercylinder area when the case in Fig. 4b is compared with the case in Fig. 3b. Due to the relatively large flow rate going through the outer cylinders as shown in Fig. 4b a strong separation behind the fourth (or last) cylinder is observed.

4.2 Elliptic cylinders

In this case, an incoming uniform flow past a slender elliptic cylinder of thickness ratio (minor to major axis) 1:6.66 at a 45^0 incidence angle is studied. Reynolds number is chosen to be Re = 200 (based on the chord length which is twice that of major axis), and the aspect ratio (the channel width over the chord length) is 20. The number of elements allocated for the stationary grid system is 14 × 16 elements in the x and y directions, and 14 × 4 elements are adopted for the fluid grid system. The detailed element layout is sketched for the first elliptic cylinder shown in Fig. 1.

When the incidence angle is 45^0 and Reynolds number is Re = 200, a well-known *Kármán* vortex street develops [13]. The streamline plots shown in Fig. 5 illustrate the history of separation behind the elliptic cylinder within a cycle. If one regards the separation starting from the leading edge as seen in Fig. 5a, the time evolution of separation is described as follows: the separation region continues to increase toward the trailing edge (Fig. 5b) and up to the trailing edge where the maximal lift holds. After the separation breaks down (Fig. 5c), it restarts from the trailing edge (Fig. 5d) and then gradually extends to the region toward the top tip (Fig. 5e), where the minimal lift occurs. The separation also splits into two parts: one is located immediately behind the ellipse, and another forms as a vortex behind the body (Fig. 5f).

The drag and lift coefficients are found to be $-0.985 \leq C_L \leq -1.500, 1.355 \leq C_D \leq 1.781$ (as seen in Fig. 6), which are qualitatively similar to the case with thickness ratio of 1:10 in [13]. The Strouhal number is 0.275 in contrast with 0.25 in the case of a thickness ratio of 1:10.

To demonstrate the capability of the multigrid-mask method in simulating the interaction among multiple objects, we add another elliptic cylinder with thickness ratio 1:4 (chord length is 60% of the first one) in the direction of incoming flow. The element layout is also sketched in Fig. 1 and the position is placed in the wake of the first elliptic cylinder. It is very common for us to experience the traction force when we park a car and another high speed car passes by to us, or when a small plane flys into the wake of a big plane, a tremendous suction force can cause a small plane to crash into the big one.

In order to prove that the traction force acting on the second elliptic cylinder is induced by the wake effect from the front one, it is rational to plot the time history of the drag coefficient at the rear one. If any negative value of drag coefficient exists, it supports our assumption. In Fig. 7, the drag and lift coefficients of both elliptic cylinders appear in the same plot. Evidently, the negative drag coefficient for the second one indeed stands and strengthens the fact that the traction force acts on the rear elliptic cylinder. Meanwhile, the drag and lift coefficients for the front elliptic cylinder also change ($1.30 \leq C_D \leq 1.828, -0.82 \leq C_L \leq -1.39$) due to the existence of the rear one. More strikingly, the Stouhal number is reduced to 0.208, which is the same as that of the rear elliptic cylinder (resonant effect), whose drag and lift coefficients are $-0.139 \leq C_D \leq 0.360, -0.939 \leq C_L \leq 0.911$, respectively.

The streamline plots as seen in Fig. 8 give a detailed description of the aforementioned traction effect. The phenomenon of the front elliptic cylinder is very similar to that of the single case; separation starts from the leading edge and grows up to the trailing edge where the separation breaks down, then restarts from the trailing edge and extends toward the leading edge where it splits into two parts, one on the surface with a small intensity and another in the wake region. The traction force can be judged based on the vortex formation on the surface of the second elliptic cylinder. Whenever the vortex formation appears on the front surface of the second one, the drag coefficient turns into a negative value as indicated in Fig. 8c. The negative value persists during the time period (Fig. 8c - Fig. 8e) when the separation on the surface of the front elliptic cylinder breaks down at the tail and restarts from the bottom and extends toward the tip. The intensity of the traction force turns out to be the strongest when the wake zone resulting from the first elliptic cylinder acts on the front surface of the second one and becomes the largest (Fig. 8d).

5 Conclusions

The solution of the Navier-Stokes equations in primitive variable form has been obtained by the pseudospectral element method via the multigrid-mask SAP domain decomposition technique. The solution procedure for flow past multiple (or single) objects includes two basic steps: a homogeneous step (mask method) and a heterogeneous step of (multigrid method). The solution on the stationary grid is first solved by the mask method, then the iterative solution between the heterogeneous step, the solution on the fluid grid, and the homogeneous (mask) step is repeated by the SAP technique with multigrid method.

The homogeneous step permits flow into the stationary grid contained in each object but subject to the restriction that flow inside or on the surface of objects should be small within the prescribed error index. The merit of the mask method is its simplicity to first provide an approximate solution of flow field by the fast eigenfunction solver. The implementation of heterogeneous step is next used to correct the flow field predicted from the homogeneous step by considering the actual contour and exact boundary conditions on the surface of objects.

From the solution point of view, the problem can be interpreted as the local fluid grid representing the objects fully overlapped with the global stationary grid standing for the entire computational domain. The SAP iterative technique bridges the data communication between the local and global coordinate systems. During the data exchange between the fluid grid (fine-grid) domain and the stationary grid (coarse-grid) domain, the coarse-grid correction technique is used to eliminate the high frequency error caused by the data interpolation from the fine-grid domain to the coarse-grid domain.

Test problems demonstrate the versatility of the proposed multigrid-mask method. Future research will concentrate on solution of flow in the three-dimensional geometries.

Acknowledgment

The work of the first author was partially supported by the Department of the Navy, Space and Naval Warfare Systems Command, under the Contract Number N00039 - 95 - C - 0002.

References

[1] M. Ingber, *Int. J. Numer. Methods Fluids* **9**, 263 (1989).

[2] T. Tran-Cong and N. Phan-Thien, *Phys. Fluids* **A 1**, 453 (1989).

[3] X. Li, H. A. Zhou, and C. Pozrikidis, *J. Fluid Mech.* **11**, 1 (1994).

[4] S. O. Unverdi and G. Tryggvason, *J. Comput. Phys.* **100**, 25 (1992).

[5] M. Briscolini and P. Santangelo, *J. Comput. Phys.* **84**, 57 (1989).

[6] C. S. Peskin, *J. Comput. Phys.* **25**, 220 (1977).

[7] H. C. Ku, R. S. Hirsh, and T. D. Taylor, *J. Comput. Phys.* **70**, 439 (1987).

[8] H. C. Ku, *J. Comput. Phys.* **117**, 215 (1995).

[9] H. C. Ku and B. Ramaswamy, to appear in *Intl. J. Numer. Methods in Eng.*

[10] H. C. Ku, R. S. Hirsh, T. D. Taylor, and A. P. Rosenberg, *J. Comput. Phys.* **83**, 260 (1989).

[11] A. J. Chorin, *Math. Comp.* **22**, 745 (1968).

[12] Y. S. Wong, T. A. Zang, and M. Y. Hussaini, *Comput. & Fluids* **14**, 85 (1986).

[13] H. J. Lugt and H. J. Haussling, *J. Fluid Mech.* **65**, 711 (1974).

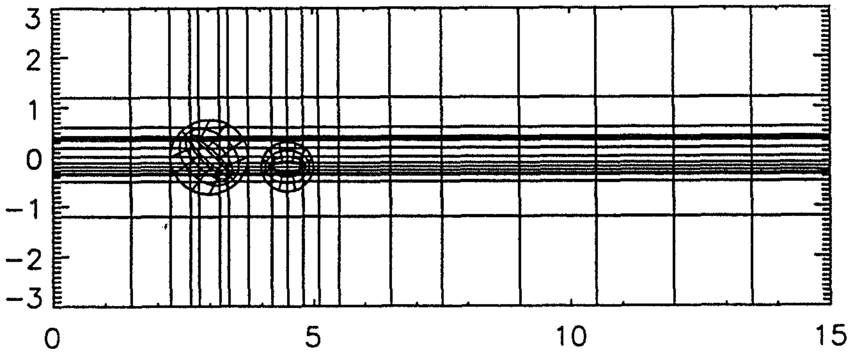


Fig. 1 Element layout for flow past elliptic cylinders

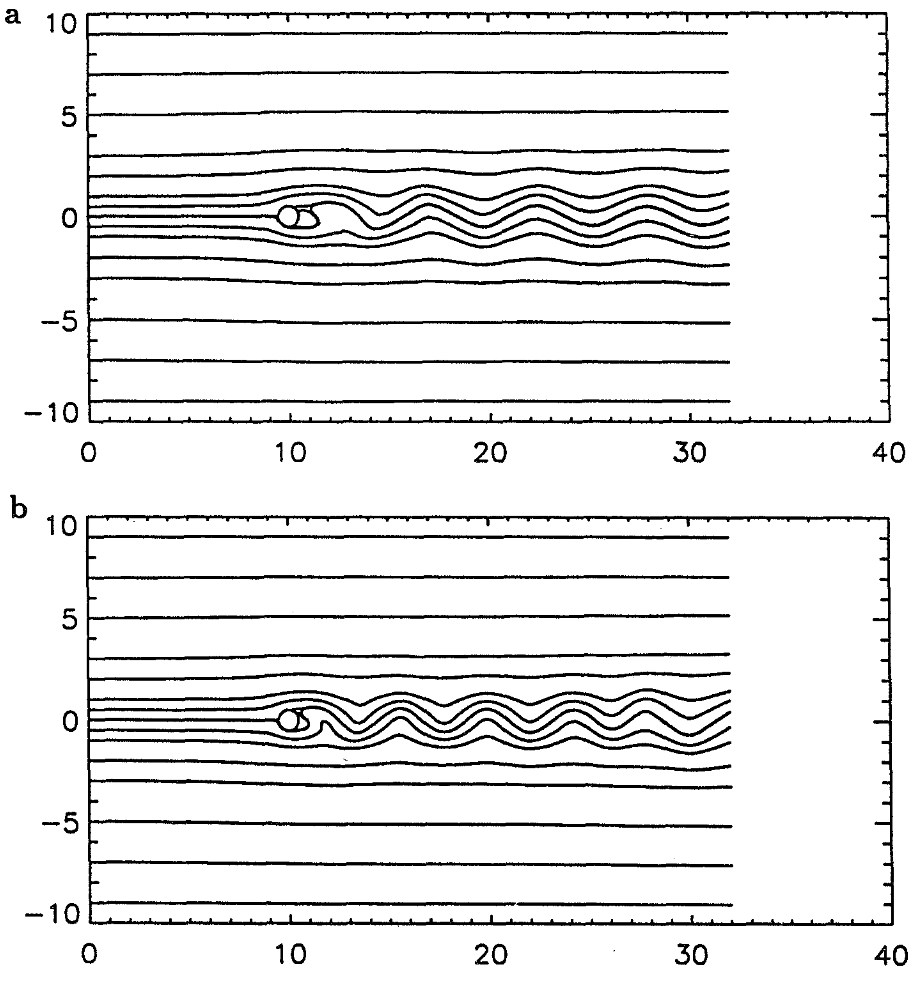


Fig. 2 Streamline plots for flow past a cylinder for (a) Re = 100, and (b) Re = 250

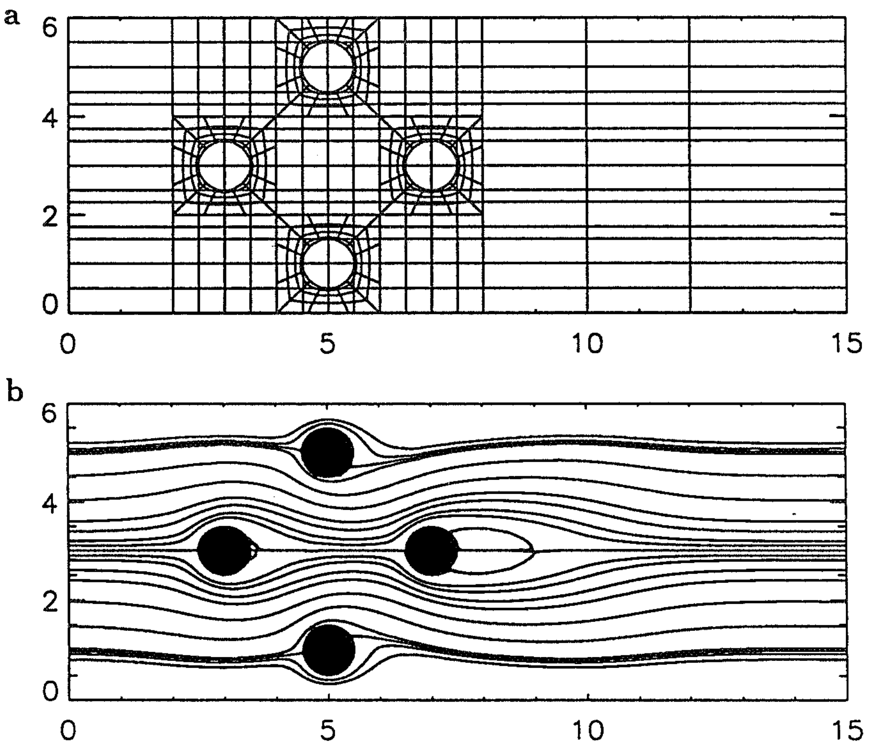


Fig. 3 Flow past four cylinders at Re = 20 with (a) element layout, and (b) streamline plot

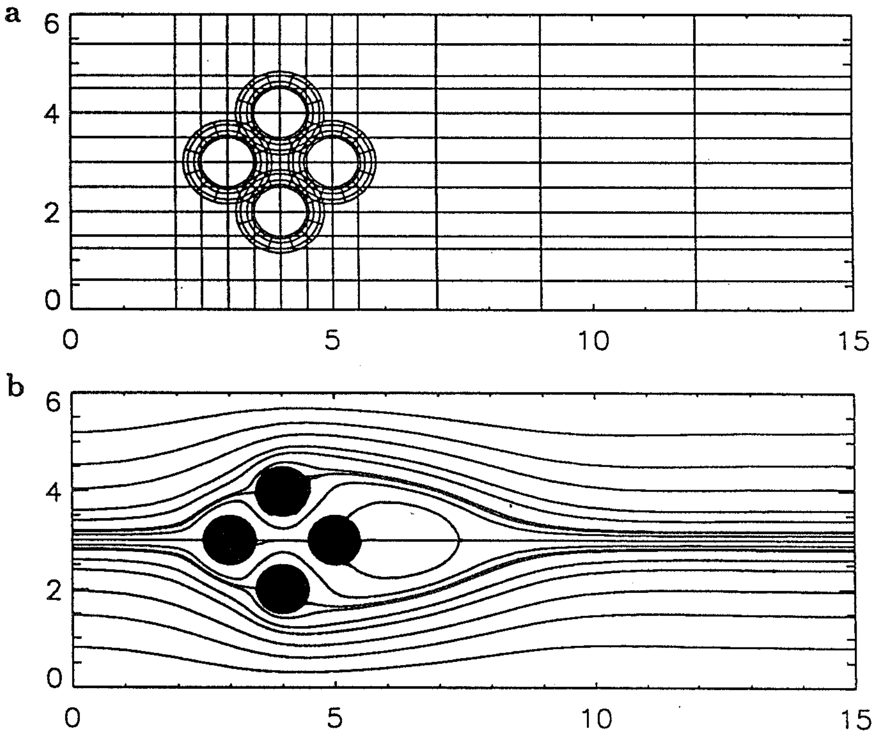


Fig. 4 Flow past four cylinders at Re = 20 with (a) element layout, and (b) streamline plot

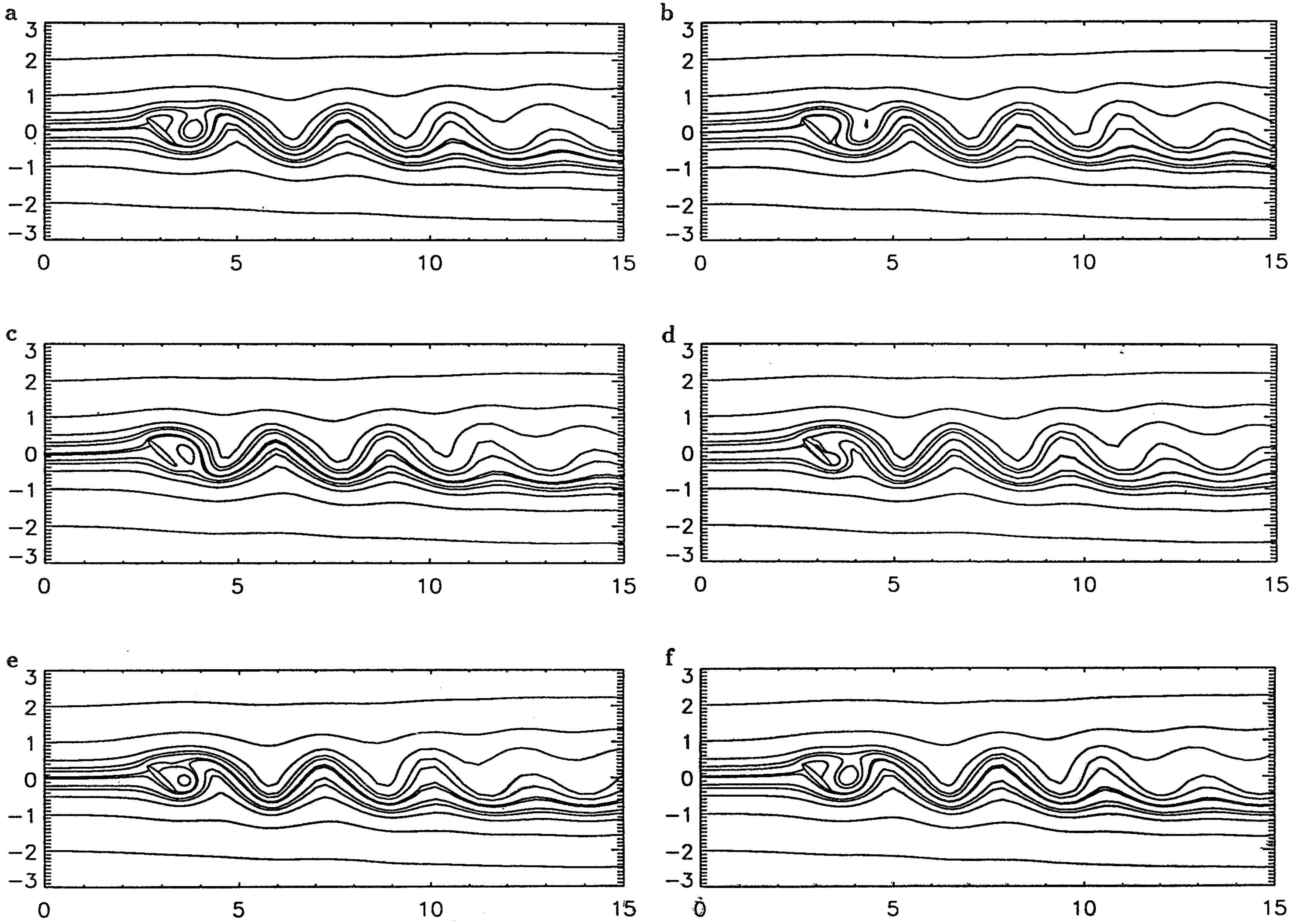


Fig. 5 Full-cycle time history of streamline plots for Re = 200 at time (a) t = 0, (b) t = 0.2T, (c) t = 0.4T, (d) t = 0.6T, (e) t = 0.8T, (f) t = T

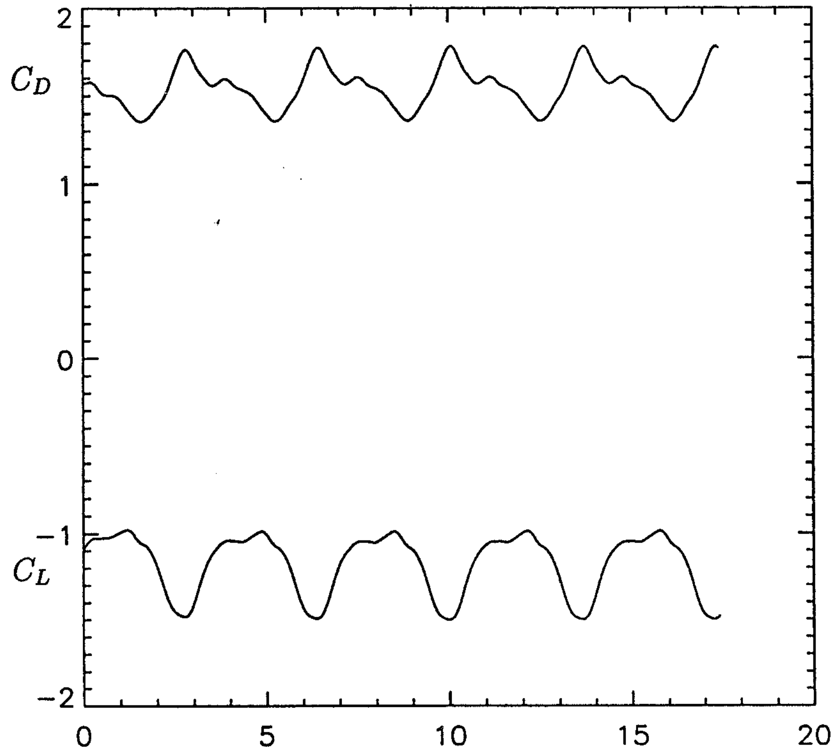


Fig. 6 Time history of drag C_D and lift C_L coefficients for flow past an elliptic cylinder at Re = 200

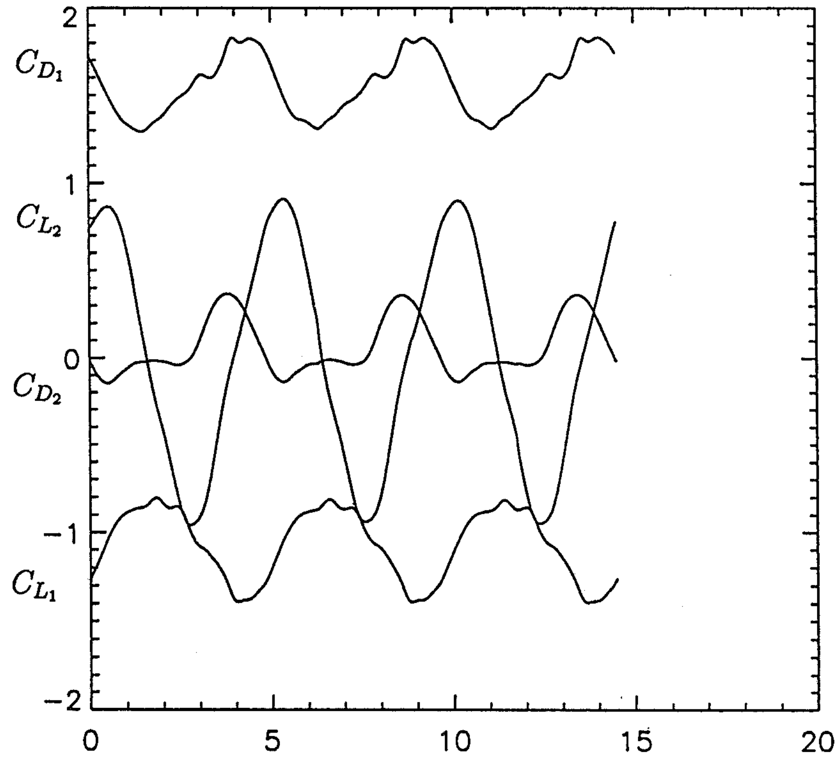


Fig. 7 Time history of drag C_D and lift C_L coefficients for flow past elliptic cylinders at Re = 200

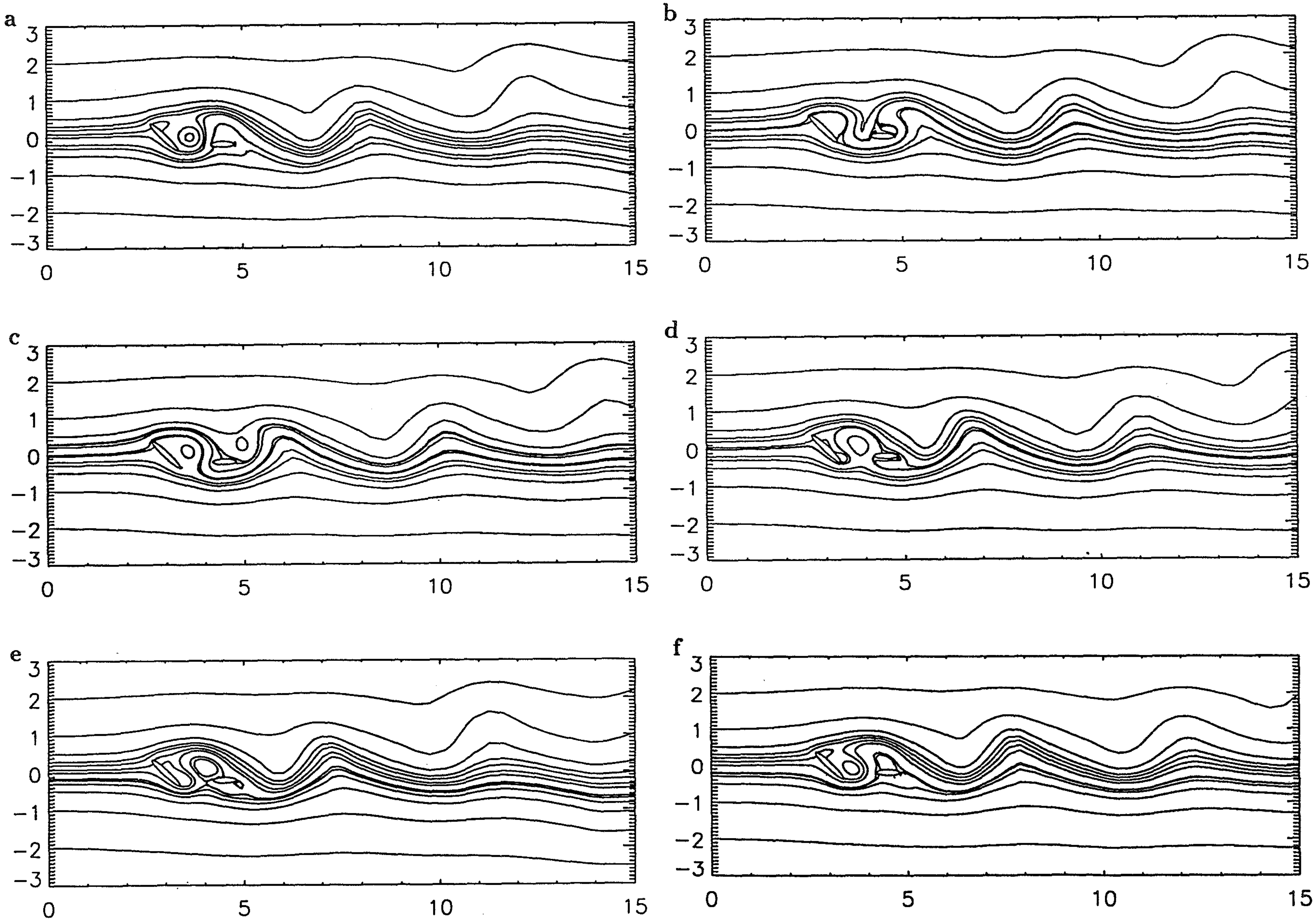


Fig. 8 Time history of streamline plots for Re = 200 at time (a) t = 0, (b) t = 0.27T, (c) t = 0.46T, (d) t = 0.67T, (e) t = 0.77T, (f) t = 0.91T

Page intentionally left blank

IMPLEMENTATION OF HYBRID V-CYCLE MULTILEVEL METHODS FOR MIXED FINITE ELEMENT SYSTEMS WITH PENALTY

Chen-Yao G. Lai
Department of Mathematics
National Chung-Cheng University
Chia-Yi, Taiwan

SUMMARY

The goal of this paper is the implementation of hybrid V-cycle hierarchical multilevel methods for the indefinite discrete systems which arise when a mixed finite element approximation is used to solve elliptic boundary value problems. By introducing a penalty parameter, the perturbed indefinite system can be reduced to a symmetric positive definite system containing the small penalty parameter for the velocity unknown alone. We stabilize the hierarchical spatial decomposition approach proposed by Cai, Goldstein, and Pasciak for the reduced system. We demonstrate that the relative condition number of the preconditioner is bounded uniformly with respect to the penalty parameter, the number of levels and possible jumps of the coefficients as long as they occur only across the edges of the coarsest elements.

INTRODUCTION

We shall be concerned with solving the discrete equations which arise when the mixed approximation is used for second order elliptic boundary value problems. Specifically, we consider the mixed approximation based on the Raviart-Thomas spaces [12]. Such approximations lead to the solution of linear systems involving block matrices of the form

$$\begin{pmatrix} M & N^T \\ N & 0 \end{pmatrix}.$$

Here M is symmetric and positive definite and N^T is the transpose of the matrix N. This matrix is clearly symmetric and indefinite.

Instead of solving this system directly, we consider solving the penalty approximation to it (cf. [1],[5]). This approximation involves the use of a small parameter ε ($10^{-3} \sim 10^{-8}$ in practice) and results in a linear system involving the block form

$$\begin{pmatrix} M & N^T \\ N & -\varepsilon I \end{pmatrix}.$$

The linear system of this form can be reduced to the solution of the matrix

$$M + \varepsilon^{-1} N N^T. \tag{1}$$

Although the matrix in (1) is symmetric and positive definite, it can have a large condition number of the order $O(\varepsilon^{-1}h^{-2})$. Here, h is the discretization parameter.

The hierarchical space decomposition method proposed in [8] reduces the above condition number to the order $O(h^{-1}\log\frac{1}{h})$. That is, the dependence of the penalty parameter ε has been removed and a reduction in the mesh dependence has been achieved. In the same paper [8], a negative result for the standard application of the multigrid method to the reduced system has been suggested. The asymptotic behavior for the standard multigrid method remains of the order $O\left(\varepsilon^{-1}h^{-2}\right)$.

In this paper, we stabilize the hierarchical spatial decomposition approach from [8] by allowing hybrid V-cycle type multilevel iterations developed by Axelsson and Vassilevski (cf. [2], [3], [13], [14]). This means that we use a pure V-cycle iteration at most of the levels while we perform a ν-fold ($\nu > 1$) cycle iteration at levels whose index is proportional to a fixed integer parameter k_0. We demonstrate that the hybrid V-cycle hierarchical multilevel preconditioners constructed in this manner give relative condition numbers that are uniformly bounded with respect to both the penalty parameter ε and the number of discretization levels if k_0 is sufficiently large and ν (the number of recursive calls at every k_0 level) satisfies certain inequalities determined only by k_0.

Finally, we note that there are other approaches suggested in Bramble, Pasciak, and Xu [6], Ewing, Lazarov, and Vassilevski [9], Mathew [11] for indefinite systems that arise in mixed finite element discretizations of second-order elliptic problems. Some of these methods are based on reducing the indefinite systems by working in divergence-free finite element spaces to obtain a system with a symmetric and positive definite matrix.

STATEMENT OF THE PROBLEM

Let Ω be a two-dimensional polygon and consider the following boundary value problem:

$$\begin{cases} -\nabla\cdot(k\nabla p) &= f, \quad \text{in } \Omega, \\ p &= 0, \quad \text{on } \Gamma = \partial\Omega, \end{cases} \tag{2}$$

where $f \in L^2(\Omega)$ and $k = k(x)$ ($x \in \Omega$ is bounded from above and below by some positive constants).

We shall use the following space to describe the corresponding variational problems. We consider the Hilbert space

$$H(\text{div};\Omega) \equiv \left\{ \mathrm{v} \in [L^2(\Omega)]^2 \ \ |\nabla\cdot\mathrm{v} \in L^2(\Omega) \right\}$$

with norm defined by

$$||\mathrm{v}||^2_{H(\text{div};\Omega)} \equiv ||\mathrm{v}||^2_{L^2(\Omega)} + ||\nabla\cdot\mathrm{v}||^2_{L^2(\Omega)}.$$

In (2), we set $\mathbf{u} = -k\nabla p$. Then we obtain the following variational equations:

$$\begin{cases} (k^{-1}\mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = 0, & \text{for all } \mathbf{v} \in H(\mathrm{div}; \Omega), \\ (\nabla \cdot \mathbf{u}, q) = (f, q), & \text{for all } q \in L^2(\Omega). \end{cases}$$

Here $(\cdot, \cdot)$ denotes the inner product in $L^2(\Omega)$ or $[L^2(\Omega)]^2$.

We assume that we are given two finite dimensional subspaces

$$V^h \subset H(\mathrm{div}; \Omega) \quad \text{and} \quad W^h \subset L^2(\Omega)$$

defined on a quasi-uniform mesh with elements of size $O(h)$. The mixed finite element approximation of $(\mathbf{u}, p)$ is then defined to be the pair, $(\mathbf{u}^h, p^h) \in \mathbf{V}^h \times W^h$, satisfying

$$\begin{cases} (k^{-1}\mathbf{u}^h, \mathbf{v}) - (p^h, \nabla \cdot \mathbf{v}) = 0, & \text{for all } \mathbf{v} \in V^h, \\ -(\nabla \cdot \mathbf{u}^h, q) = -(f, q), & \text{for all } q \in W^h. \end{cases} \tag{3}$$

Problem (3) can be reformulated in terms of operators. We define operators $M : V^h \to V^h$, $N : V^h \to W^h$ and $N^* : W^h \to V^h$ by

$$\begin{aligned} (M\mathbf{v}, \boldsymbol{\psi}) &\equiv (k^{-1}\mathbf{v}, \boldsymbol{\psi}), && \text{for all } \boldsymbol{\psi} \in V^h, \\ (N\mathbf{v}, q) &\equiv -(\nabla \cdot \mathbf{v}, q), && \text{for all } q \in W^h, \\ (N^*q, \mathbf{v}) &\equiv -(q, \nabla \cdot \mathbf{v}), && \text{for all } \mathbf{v} \in V^h. \end{aligned}$$

With this notation, (3) takes the following form:

$$\begin{pmatrix} M & N^* \\ N & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^h \\ p^h \end{pmatrix} = \begin{pmatrix} 0 \\ -f^h \end{pmatrix}, \tag{4}$$

where f^h denotes the $L^2(\Omega)$ orthogonal projection of f onto W^h.

The solution $(\mathbf{u}^h, p^h)$ can be approximated by regularization (i.e., by solving a reduced system using a penalty approximation). Let $\varepsilon > 0$ be a small (penalty) parameter. We consider the solution of the following perturbed system:

$$\begin{pmatrix} M & N^* \\ N & -\varepsilon I \end{pmatrix} \begin{pmatrix} \mathbf{u}_\varepsilon^h \\ p_\varepsilon^h \end{pmatrix} = \begin{pmatrix} 0 \\ -f^h \end{pmatrix}. \tag{5}$$

Eliminating p_ε^h in (5) gives rise to the following reduced problem for $\mathbf{u}_\varepsilon^h$:

$$A^\varepsilon \mathbf{u}_\varepsilon^h \equiv \left(M + \frac{1}{\varepsilon} N^* N\right) \mathbf{u}_\varepsilon^h = -\varepsilon^{-1} N^* f^h. \tag{6}$$

The operator A^ε is obviously symmetric and positive definite. We note that once $\mathbf{u}_\varepsilon^h$ has been determined from (6), p_ε^h can be computed by

$$p_\varepsilon^h = \varepsilon^{-1} \left(N\mathbf{u}_\varepsilon^h + f^h\right).$$

The penalty method was analyzed in [1] and [5] for a class of mixed approximations. It follows from these results that, for the Raviart-Thomas spaces [12],

$$||\mathbf{u} - \mathbf{u}_\varepsilon^h||_{H(\mathrm{div};\Omega)} + ||p - p_\varepsilon^h||_{L^2(\Omega)}$$

$$\leq C \left[\inf_{\mathbf{v} \in V^h} ||\mathbf{u} - \mathbf{v}||_{H(\mathrm{div};\Omega)} + \inf_{q \in W^h} ||p - q||_{L^2(\Omega)} + \varepsilon ||p||_{L^2(\Omega)} \right],$$

where the constant C is independent of both ε and h.

Moreover, we note that the problem (4) is indefinite and of saddle-point type. An adequate approximation can be provided if the finite element spaces V^h and W^h satisfy the Babuška-Brezzi stability condition (cf. Babuška [4] and Brezzi [7]). This means that for some positive constant β independent of the mesh parameter h the following stability condition holds:

$$\sup_{\mathbf{v} \in V^h} \frac{(\nabla \cdot \mathbf{v}, q)}{||\mathbf{v}||_{H(\mathrm{div};\Omega)}} \geq \beta ||q||_0^2 \qquad \text{for all } q \in W^h.$$

In the remainder of this section, we describe the Raviart-Thomas spaces on the triangle T. The Raviart-Thomas space of order r (a given nonnegative integer) on the triangle T for the velocity is defined by

$$\mathbf{V}^h(T) = \left\{ \mathbf{v} \in [P_r(T)]^2 \oplus \mathbf{v}_0 \right\},$$

where

$$\mathbf{v}_0 = \begin{pmatrix} x_1 \hat{P}_r(x) \\ x_2 \hat{P}_r(x) \end{pmatrix}$$

and $\hat{P}_r(x)$ is a homogeneous polynomial of degree r. The corresponding space for the pressure is given by

$$W^h(T) = P_r(T),$$

where $P_r(T)$ is a polynomial of degree r defined on the triangle T. We also consider the projection operator π_h that is defined by the following:

$$\begin{cases} (\pi_h \mathbf{v} \cdot n, q)_E &= (\mathbf{v} \cdot n, q)_E, \quad \text{for } q \in P_r(E) \text{ and all three edges } E \text{ of } T, \\ (\pi_h \mathbf{v}, \boldsymbol{\psi})_T &= (\mathbf{v}, \boldsymbol{\psi})_T, \quad \text{for } \boldsymbol{\psi} \in (P_{r-1}(T))^2. \end{cases} \tag{7}$$

HIERARCHICAL SPACE DECOMPOSITION METHOD

In this section, we shall describe the hierarchical spatial decomposition method [8]. We start with a coarse initial triangulation $\mathcal{T}_0$ of the domain Ω. For any element $T \in \mathcal{T}_0$, we consider the local ellipticity constants

$$\sigma_T = \frac{\sup\limits_{x \in T} k(x)}{\inf\limits_{x \in T} k(x)}$$

and

$$\sigma = \max_{T\in\mathcal{T}_0} \sigma_T.$$

Note that σ can remain close to 1 even when the coefficient $k(x)$ has large jumps, as long as these occur only across edges of elements from $\mathcal{T}_0$.

We next construct a nested family of triangulations

$$\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_J \equiv \mathcal{T}_h$$

of the domain Ω by subdividing each element of $\mathcal{T}_j$ into four congruent ones to obtain $\mathcal{T}_{j+1}$. We consider the Raviart-Thomas velocity space V_j for every triangulation $\mathcal{T}_j$ (with mesh size $h_j = 2^{-j}h_0$). For each level $j = 1, 2, \cdots, J$, we let

$$\pi_j \mathbf{v} = \pi_{h_j}\mathbf{v},$$

where π_{h_j} is the projection operator defined in (7). For convenience, we shall let $\pi_{-1} = 0$.

We define the spaces M_j to be the images of the operators $(\pi_j - \pi_{j-1})$ acting on elements from V_h

$$M_j = \{\mathbf{w} = (\pi_j - \pi_{j-1})\mathbf{v}, \quad \text{all } \mathbf{v} \in V_h\}.$$

It is clear that $\{M_j\}$ are subspaces of $V_J \equiv V_h$ satisfying

$$V_j = M_0 \oplus M_1 \oplus \ldots \oplus M_j, \quad j = 0, 1, \ldots J.$$

For $j = 0, 1, \ldots J$, we define the operator A_j to be

$$(A_j v, w) \equiv A^\epsilon(v, w), \quad \text{for all } v, w \in V_j.$$

We next define the operators D_j to be $A^\epsilon_{M_j}$. That is,

$$(D_j\boldsymbol{\psi}, \boldsymbol{\theta}) = A^\epsilon(\boldsymbol{\psi}, \boldsymbol{\theta}), \quad \text{for all } \boldsymbol{\psi}, \boldsymbol{\theta} \in M_j,$$

where $\boldsymbol{\psi} = (\pi_j - \pi_{j-1})\mathbf{v}$ and $\boldsymbol{\theta} = (\pi_j - \pi_{j-1})\mathbf{w}$ for some $\mathbf{v}, \mathbf{w} \in V_J$.

The primitive form of the hierarchical preconditioner proposed in [8] can be written as

$$(B_J^H v, w) = (B_0\pi_0 v, \pi_0 w) + \sum_{\sigma=1}^{J} (D_\sigma \psi_\sigma, \theta_\sigma),$$

where $B_0 = A_0$, $\psi_\sigma = (\pi_\sigma - \pi_{\sigma-1})v$, and $\theta_\sigma = (\pi_\sigma - \pi_{\sigma-1})w$. To obtain an efficient preconditioner $\tilde{D}_j$ for D_j (cf. [8]), we use the decomposition

$$\boldsymbol{\psi} = \boldsymbol{\psi}_H + \boldsymbol{\psi}_P,$$

where $\boldsymbol{\psi}_H \in M_j$ is defined element-wise for any $T \in \mathcal{T}_{j-1}$ such that

$$\begin{cases} A^\varepsilon(\boldsymbol{\psi}_H, \boldsymbol{\theta}) &= 0, \qquad \text{for all } \boldsymbol{\theta} \in M_j, \text{ and } \theta \cdot n = 0 \text{ on } \partial T; \\ \boldsymbol{\psi}_H \cdot n\Big|_{\partial T} &= \boldsymbol{\psi} \cdot n\Big|_{\partial T}. \end{cases}$$

Let $\tilde{D}_j^H$ be an appropriately scaled diagonal part of D_j^H such as

$$\left(\tilde{D}_j^H \mathrm{v}, \mathrm{v}\right) = C \sum_{\substack{\text{all edges } E \\ \text{of all } T \in \mathcal{T}_{i-1}}} h_{j-1}^2 k_E \sum_{\substack{\{x_s\}_{s=0}^r \text{ a set} \\ \text{of nodes on } E}} (\mathrm{v} \cdot n|_E)^2 (x_s)$$

for some constant $C > 0$ independent of h_{j-1} and for some weights

$$k_E = \frac{1}{\max\limits_{T_1} k} + \frac{1}{\max\limits_{T_2} k}, \qquad \text{where } \overline{T}_1 \cap \overline{T}_2 = E \text{ and } T_1, T_2 \in \mathcal{T}_{j-1}.$$

Then we can write $\tilde{D}_j$ as

$$\left(\tilde{D}_j \boldsymbol{\psi}, \boldsymbol{\chi}\right) \equiv \left(\tilde{D}_j^H \boldsymbol{\psi}_H, \boldsymbol{\chi}_H\right) + \left(D_j^P \boldsymbol{\psi}_P, \boldsymbol{\chi}_P\right).$$

The final form for the hierarchical preconditioner becomes

$$(\tilde{B}_J^H v, w) = (B_0 \pi_0 v, \pi_0 w) + \sum_{\sigma=1}^{J} \left(\tilde{D}_\sigma \psi_\sigma, \theta_\sigma\right). \tag{8}$$

We now state the following theorem for the hierarchical preconditioner [8] without proof.

Theorem 1. *For any vector function* $\mathrm{v} \in V_J$*, we have that*

$$C 2^{-J} \tilde{B}_J^H(\mathrm{v}, \mathrm{v}) \leq A^\varepsilon(\mathrm{v}, \mathrm{v}) \leq C J \tilde{B}_J^H(\mathrm{v}, \mathrm{v}),$$

where C *is a constant independent of* ε*,* J*, and the mesh size.*

The above theorem shows that the hierarchical preconditioner can be used to effectively precondition the original form A^ε as long as J is not too large.

HYBRID V-CYCLE MULTILEVEL PRECONDITIONERS

We shall describe the hybrid V-cycle multilevel preconditioners in this section. The construction of these multilevel preconditioners is based on the hierarchical preconditioner (8) and some polynomial acceleration techniques proposed in [2], [3], [13], and [14]. The purpose of the polynomial acceleration is to stabilize the growth of the condition number for the hierarchical preconditioner. The hybrid V-cycle multilevel preconditioner B_j is defined by recursion as follows.

- Let $p_\nu(t)$ be a given polynomial of degree $\nu \geq 1$ such that

$$\begin{cases} \text{(i) } p_\nu(0) = 1, \\ \\ \text{(ii) } 0 \leq p_\nu(t) < 1 \quad \text{for } t \in (0, 1]. \end{cases}$$

- For a given integer parameter $k_0 \geq 1$, we set

  1. $B_0 = A_0$,
  2. $(B_{k_0}v, w) = (B_0\pi_0 v, \pi_0 w) + \sum_{\sigma=1}^{k_0} \left(\tilde{D}_\sigma \psi_\sigma, \theta_\sigma\right)$.

- For $s = 1, 2, \ldots$, $m = sk_0$, and for all j such that $m \leq j \leq m + k_0$, we first define operator B_m to be

$$(B_m v, w) = (B_0\pi_0 v, \pi_0 w) + \sum_{\sigma=1}^{m} \left(\tilde{D}_\sigma \psi_\sigma, \theta_\sigma\right) \quad (\forall v, w \in V_m).$$

  Then the operator B_j is obtained for all $v, w \in V_j$ by the relation

$$(B_j v, w) = \left(\tilde{B}_m \pi_m v, \pi_m w\right) + \sum_{\sigma=m+1}^{j} \left(\tilde{D}_\sigma(\pi_\sigma v - \pi_{\sigma-1} v), (\pi_\sigma w - \pi_{\sigma-1})w\right).$$

We next present some technical lemmas which are used to prove the convergence results for the hybrid V-cycle multilevel preconditioners. We will state these lemmas without proof. We refer to [10] for detailed proof.

Lemma 1. *For any function* $\mathrm{v} \in V_{j+k_0}$*, we have that*

$$A^\varepsilon(\pi_j \mathrm{v}, \pi_j \mathrm{v}) \leq \eta(k_0) A^\varepsilon(\mathrm{v}, \mathrm{v}),$$

where $\eta(k_0) = C2^{k_0}$ *and the constant* C *is independent of* j *and the penalty parameter* ε.

Lemma 2. *Let* m *and* j $(> m)$ *be given integers. The following inequality holds for some constant* $\delta_m \geq 0$*:*

$$(A_m v, v) \leq \left(\tilde{B}_m v, v\right) \leq (1 + \delta_m)(A_m v, v), \qquad \text{for all } v \in V_m.$$

Lemma 3. *The following spectral equivalence relation holds for all* $v \in V_j$*:*

$$\frac{1}{j - m + 1}(A_j v, v) \ \leq\ (B_j v, v) \ \leq\ \Big(\delta_m \eta(j - m) + \eta(j - m) + b\eta(1) \sum_{\sigma=m+1}^{j} \eta(j - \sigma)\Big)(A_j v, v).$$

We shall use the following polynomial $p_\nu(t)$ for the preconditioner:

$$p_\nu(t) = \frac{1 + T_\nu\left(\frac{1+\alpha-2t}{1-\alpha}\right)}{1 + T_\nu\left(\frac{1+\alpha}{1-\alpha}\right)}$$

with

$$\nu > \sqrt{(k_0+1)\eta(k_0)}. \tag{9}$$

Here T_ν is the Chebyshev polynomial of degree ν.

Let α be a small positive parameter satisfying

$$\sup\left\{\frac{1}{1-p_\nu(t)}, t \in [\tilde{\alpha}, 1]\right\} = \left[\frac{\left(1-\sqrt{\tilde{\alpha}}\right)^\nu + \left(1+\sqrt{\tilde{\alpha}}\right)^\nu}{2\sqrt{\tilde{\alpha}}\sum_{\sigma=1}^{\nu}\left(1-\sqrt{\tilde{\alpha}}\right)^{\nu-\sigma}\left(1+\sqrt{\tilde{\alpha}}\right)^{\sigma-1}}\right]^2,$$

where the parameter $\tilde{\alpha} = \dfrac{\alpha}{k_0+1}$.

We note that such a parameter exists under the above choice of ν because in this case we have the following asymptotic relation:

$$\left[\frac{\left(1-\sqrt{\tilde{\alpha}}\right)^\nu + \left(1+\sqrt{\tilde{\alpha}}\right)^\nu}{2\sqrt{\tilde{\alpha}}\sum_{\sigma=1}^{\nu}\left(1-\sqrt{\tilde{\alpha}}\right)^{\nu-\sigma}\left(1+\sqrt{\tilde{\alpha}}\right)^{\sigma-1}}\right]^2 \sim \frac{1}{\nu^2\tilde{\alpha}}$$

and for a sufficiently small α we solve the inequality for ν

$$\frac{1}{\nu^2\tilde{\alpha}} < \frac{1}{\alpha\eta(k_0)}.$$

Let λ_j be the largest eigenvalue of $A_j^{-1}B_j$. An upper bound for λ_{m+k_0} is given as follows.

Lemma 4.

$$\lambda_{m+k_0} \leq \eta(k_0)\sup\left\{\frac{1}{1-p_v(t)}, t \in \left[\frac{\alpha}{k_0+1}, 1\right]\right\} + b\eta(1)\sum_{\sigma=1}^{k_0}\eta(\sigma) \leq \frac{1}{\alpha}. \tag{10}$$

The multilevel preconditioner B_j will be spectrally equivalent to A_j. We summarize these results in the following theorem.

Theorem 2. *Let ν satisfy the inequality (9) for some given integer parameter $k_0 \geq 1$. For $\alpha \in (0,1]$ that is sufficiently small and satisfies the inequality (10), the following spectral relation between B_j and A_j holds uniformly for $j \geq 0$:*

$$\frac{1}{k_0+1}(A_jv,v) \leq (B_jv,v) \leq \frac{1}{\alpha}(A_jv,v) \quad \textit{for all } v \in V_j.$$

COMPUTATIONAL COMPLEXITY OF THE PRECONDITIONER

To study the computational costs, we denote the degrees of freedom at level j by n_j. From the triangulation process, we may assume that $n_{j+1}/n_j = 4$. Let $\mathcal{W}_{s+1}$ be the number of arithmetic operations performed at level $(s+1)k_0$. We then obtain the following recursive formula:

$$\mathcal{W}_{s+1} = Cn_{(s+1)k_0} + \nu\mathcal{W}_s, \tag{11}$$

where the second term on the right-hand-side stands for ν recursive calls of the preconditioner B_{sk_0} (the polynomial corrections at level sk_0). Thus, the computation of this action is

$$\tilde{B}^{-1}_{sk_0} = \left[I - p_\nu\left(\frac{1}{k_0+1}B^{-1}_{sk_0}A_{sk_0}\right)\right]A^{-1}_{sk_0}. \tag{12}$$

We note that the first term on the right-hand side of (11) stands for the work to invert the block-diagonal matrices $\tilde{D}^H_j$ and D^P_j and ν actions of the matrix A_{sk_0} involved in (12). Thus, in general, C is a function of the parameter ν and k_0 ($C = C(\nu, k_0)$). To obtain an optimal order preconditioner in terms of computational complexity, we choose ν and k_0 such that

$$\mathcal{W}_{s+1} \leq \text{const } n_{(s+1)k_0}.$$

Using (11) recursively, we obtain

$$\begin{aligned} \mathcal{W}_{s+1} &= C\left(n_{(s+1)k_0} + \nu n_{sk_0} + \ldots + \nu^s n_{k_0}\right) + \nu^{s+1}\mathcal{W}_0 \\ &= Cn_{(s+1)k_0}\left[\left(\frac{\nu}{2^{2k_0}}\right)^{s+1}\frac{\mathcal{W}_0}{n_0} + \sum_{\sigma=0}^{s}\left(\frac{\nu}{2^{2k_0}}\right)^{\sigma}\right]. \end{aligned}$$

Hence the condition for an optimal order preconditioner is

$$\frac{\nu}{2^{2k_0}} < 1.$$

This is the constraint for determining the parameters ν and k_0 to be an optimal hybrid V-cycle multilevel preconditioner.

In order to make B_j spectrally equivalent to A_j as given in Theorem 2, we need to impose another constraint for choosing parameters ν and k_0 as follows:

$$\nu > \sqrt{(k_0+1)\eta(k_0)}.$$

Therefore, we establish the following relation for the parameters ν and k_0 to guarantee both the optimality and the spectrally equivalent property for the hybrid V-cycle multilevel preconditioner. The relation reads as follows:

$$2^{2k_0} > \nu > C\sqrt{\sigma(k_0+1)2^{k_0}}. \tag{13}$$

These relations can always be satisfied because k_0 can be sufficiently large. We summarize the above results in the next theorem.

Theorem 3. *The hybrid V-cycle preconditioner B_j with the parameters specified above gives an optimal order CG method if ν and k_0 satisfy the inequalities* (13).

IMPLEMENTATION OF THE PRECONDITIONER

We first consider the hybrid V-cycle multilevel preconditioner in following matrix form:

(1) For $k = 1$, set

$$M^{(1)} = A^{(1)}.$$

(2) For $k = 2$ to J, we define

$$M^{(k)} = \begin{bmatrix} A_{11}^{(k)} & 0 \\ A_{21}^{(k)} & \widetilde{M}^{(k-1)} \end{bmatrix} \begin{bmatrix} I & A_{11}^{(k)^{-1}} A_{12}^{(k)} \\ 0 & I \end{bmatrix},$$

where

$$\begin{cases} \widetilde{M}^{(k-1)} &= M^{(k-1)}, \ k \neq sk_0 + 1; \\ (\widetilde{M}^{(k-1)})^{-1} &= \left[I - p_\nu(M^{(k-1)^{-1}} A^{(k-1)}) A^{(k-1)^{-1}}\right], \\ & \quad k = sk_0 + 1, \quad s = 1, 2, \cdots, J/k_0 - 1. \end{cases} \tag{14}$$

Here, $p_\nu(t)$ is a polynomial of degree $\nu \geq 1$ such that $p_\nu(0) = 1$ and $0 \leq p_\nu(t) < 1$, $t \in (0, 1]$.

To solve systems defined by $M = M^{(J)}$, we use the following multilevel iteration (AMLI) from [3]. Let $p_\nu^{(s)}, s = 1, 2, \cdots, J$, be given polynomials of degree ν such that $p_\nu^{(s)}(0) = 1$. Let polynomial $Q_\nu^{(s)}$ of degree $\nu - 1$ be

$$Q_\nu^{(s)} = (1 - p_\nu^{(s)})t^{-1} = q_0^{(s)} + q_1^{(s)} t + \cdots + q_{\nu-1}^{(s)} t^{\nu-1}, \ \nu = \nu^{(s)}.$$

For a given vector $\mathbf{d} = \mathbf{d}^{(J)}$, the AMLI gives

$$\begin{aligned} \mathbf{c}^{(J)} &= Q_\nu^{(J)}(M^{(J)^{-1}} A^{(J)}) M^{(J)^{-1}} \mathbf{d}^{(J)} \\ &= \left[I - p_\nu(M^{(J)^{-1}} A^{(J)})\right] A^{(J)^{-1}} \mathbf{d}^{(J)}. \end{aligned}$$

In particular, for the case $\nu_J = 1$ (i.e., $p_\nu^{(J)}(t) = 1 - t$), we have simply

$$M^{(J)^{-1}} \mathbf{d}^{(J)} = \mathbf{c}^{(J)}.$$

Algorithm AMLI. Given a set of polynomials

$$\left\{p_\nu^{(s)}(t), s = 1, 2, \cdots, J\right\}$$

such that $p_\nu^{(s)}(0) = 1$, we set

$$Q_\nu^{(s)} = q_0^{(s)} + q_1^{(s)}t + \cdots + q_{\nu-1}^{(s)}t^{\nu-1}, \quad \nu = \nu_s.$$

Then, for any vector $\mathbf{d} = \mathbf{d}^{(J)}$, the AMLI gives

$$\mathbf{c} \;=\; \left[I - p_\nu({M^{(J)}}^{-1}A^{(J)})\right] {A^{(J)}}^{-1}\mathbf{d}^{(J)}$$

in the following steps.

```
(0) *initiate*
        for $k = 1$ to $J$ set $\sigma(k) = 0$;
        $k = J$;
(1) $\sigma(k) = \sigma(k) + 1$;
        if $\sigma(k) = 1$ then
            $\mathbf{v}^{(k)} = 0,\ \mathbf{W} = q_{\nu_k-1}^{(k)}\mathbf{d}^{(k)}$;
        else
            $\mathbf{W} = q_{\nu_k-\sigma(k)}^{(k)}\mathbf{d}^{(k)} + A^{(k)}\mathbf{v}^{(k)}$;
(2) $\mathbf{v}_1^{(k)} = {A_{11}^{(k)}}^{-1}\mathbf{W}_1$;
(3) $\mathbf{d}^{(k-1)} = \mathbf{W}_2 - A_{21}^{(k)}\mathbf{v}_1$;
(4) $k := k - 1$
        if $k > 0$ go to (1);
(5) *solve on the initial level*
        $\mathbf{v}^{(1)} = Q_{\nu_1-1}^{(1)}(1){A^{(1)}}^{-1}\mathbf{d}^{(1)}$;
(6) *set*
        $\mathbf{v}_2^{(k+1)} = \mathbf{v}^{(k)}$;
(7) $\mathbf{v}_1^{(k+1)} = \mathbf{v}_1^{(k+1)} - {A_{11}^{(k+1)}}^{-1}A_{12}^{(k+1)}\mathbf{v}_2^{(k+1)}$;
(8) $k := k + 1$;
        if $\sigma(k) < \nu_k$ go to (1);
(9) $\sigma(k) = 0$;
        if $k < J$ go to (6);
(10) $\mathbf{c}^{(J)} = \mathbf{d}^{(J)}$.
```

NUMERICAL RESULTS

We present numerical results of the hybrid V-cycle multilevel preconditioners for the following two-dimensional discontinuous coefficients problem on the unit box $\Omega=(0,1)\times(0,1)$. In all experiments, the lowest order Raviart-Thomas triangular element is used. We consider the model problem given in (2), where the diffusion coefficients are assumed to be piecewise constants on the coarsest grid triangles. As a consequence, both local and global elliptic constants σ_T and σ are 1. In particular, we give the numerical results for the 32-subdomain case with k^{-1} in each subdomain as shown in Figure 1.

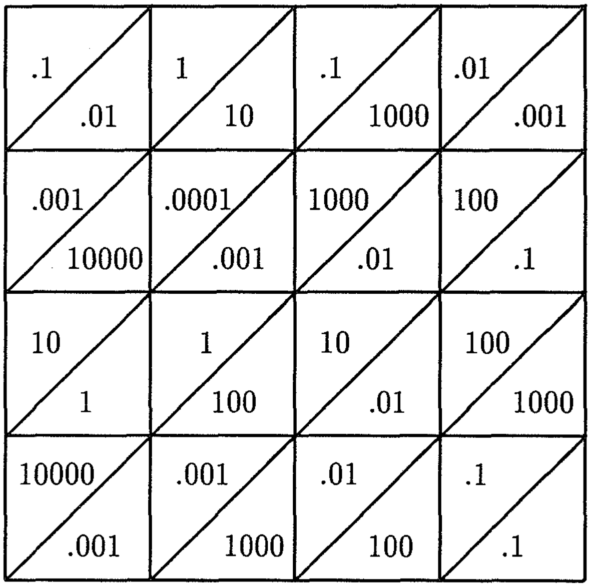


Figure 1: the coefficient k^{-1} on each subdomain

For each preconditioning step, we note that a set of polynomials of degree $\nu \equiv \nu_s$

$$Q_\nu^{(s)} = q_0^{(s)} + q_1^{(s)}t + \cdots + q_\nu^{(s)}t^\nu, \quad s = 1,2,\cdots,J$$

is used in the AMLI algorithm. These polynomials are specified by the following set of positive integers:

$$\{\nu_1, \nu_2, \nu_3, \cdots, \nu_J\},$$

which are the degree of the polynomials for each level. Here level 1 and level $J(\equiv 6)$ are the coarsest level ($h_0 = 1/4$) and finest level ($h = 1/128$), respectively. We note that ν_1 is always chosen to be one, and that the coarsest grid problem is always solved by the CG method to the machine precision ϵ_{mach}.

During the preprocessing stage, for $k = 2,3,\cdots,6$, we first estimate the extreme eigenvalues of $[M_{k-1}^{-1}A_{k-1}]$ by PCG iterations (the convergence criterion is that the reduction of the energy norm for residuals is not less than or equal to 10^{-6}).

Suppose that

$$\lambda[M_{k-1}^{-1}A_{k-1}] \subset [c,d],$$

for some constant c and d. Then the polynomial $Q_\nu^{(k)}$ is computed by the formula

$$Q_\nu^{(k)} = \frac{1-p_\nu(t)}{t},$$

where

$$\begin{cases} p_1(t) = 1-t; \\ p_\nu(t) = \dfrac{1+T_\nu[(c+d-2t)/(d-c)]}{1+T_\nu[(c+d)/(d-c)]}, \quad \nu = 2,3,\cdots. \end{cases}$$

We see that this step can be done by table lookup since ν is a small number ($\nu \in \{2,3\}$) in practice.

We refer to the set of polynomials by the set of degrees

$$\{\nu_1 = 1, \nu_2 = \nu, \nu_3 = \nu, \cdots, \nu_J = 1\}.$$

We perform numerical experiments for the following cases

(a) $(1,1,1,1,1,1)$,
(b) $(1,1,2,1,1,1)$,
(c) $(1,1,3,1,1,1)$,
(d) $(1,1,2,1,2,1)$,
(e) $(1,1,3,1,3,1)$,
(f) $(1,2,1,2,1,1)$,
(g) $(1,3,1,3,1,1)$,
(h) $(1,2,2,2,2,1)$.

All experiments were performed in one of the research computing facilities at the National Chung Cheng University. The LINPACK benchmark of the machine is about 22 mflops and the machine constant $\epsilon_{\text{mach}} \in (10^{-15}, 10^{-16})$. We measure the CPU time for both the preprocessing stage and the PCG iteration stage. We note that there is no preprocessing time for case (a) since it corresponds to the pure V-cycle hierarchical method [8].

We perform each case 5 times on the same machine and get the average time and condition numbers. The results are given in Table 1. We note that the set of polynomials (h) gives the best result for the condition number, although it is the most expensive. In addition to case (h), both (d) and (e) give very good results for the condition number. Also, most cases are less expensive than the pure V-cycle in terms of computing cost.

In Table 2, we present the results for the V-cycle and case (d). The results show that both cases have uniform condition numbers and computing times independent of the penalty parameter ε.

	$\{\nu_i\}$	preprocessing	PCG iteration	total time	κ
(a)	(1,1,1,1,1,1)	0	34.33	34.33	82.3
(b)	(1,1,2,1,1,1)	.64	28.90	29.54	45.3
(c)	(1,1,3,1,1,1)	.64	29.98	30.62	39.5
(d)	(1,1,2,1,2,1)	3.66	25.49	29.15	17.5
(e)	(1,1,3,1,3,1)	4.18	32.03	36.21	12.5
(f)	(1,2,1,2,1,1)	1.89	26.50	28.39	28.1
(g)	(1,3,1,3,1,1)	2.65	33.23	35.88	22.9
(h)	(1,2,2,2,2,1)	8.01	32.09	40.10	10.5

Table 1: computing time and condition number κ

	ε=.001		ε=.0001		ε=.00001	
$(\nu_1, \nu_2, \nu_3, \nu_4, \nu_5, \nu_6)$	κ	CPU time	κ	CPU time	κ	CPU time
(1,1,1,1,1,1)	85.3	36.37	84.7	34.33	82.9	34.46
(1,1,2,1,2,1)	18.1	27.91	17.5	25.40	17.8	26.14

Table 2: comparisons of (1,1,1,1,1,1) and (1,1,2,1,2,1) for various ε

However, the condition number for the case (d) is independent of the number of levels (there are currently six levels) while the condition number of the V-cycle does grow with the order $O(h^{-1} \log \frac{1}{h})$ (cf. [8]). Also, the computing cost for the case (d) is quite small compared to that required for the V-cycle.

CONCLUSIONS

Based on the idea of a hierarchical block preconditioner proposed by Cai, Goldstein, and Pasciak [8], we develop hybrid V-cycle multilevel preconditioners that give relative condition numbers that are uniformly bounded with respect to both the penalty parameter ε and the number of discretization levels J if we choose proper values for k_0 and ν. The numerical results confirm the uniform convergence behavior for the hybrid V-cycle multilevel preconditioners.

ACKNOWLEDGMENTS

The author would like to thank Professor P. S. Vassilevski for valuable discussions and for his continuous support.

REFERENCES

[1] Axelsson, O.: Preconditioning of indefinite problems by regularization. *SIAM J. Numer. Anal.*, vol. 16, 1979, pp. 58–69.

[2] Axelsson, O.; Vassilevski, P.S.: Algebraic multilevel methods I. *Numer. Math.*, vol. 56, 1989, pp. 157–177.

[3] Axelsson, O.; Vassilevski P.S.: Algebraic multilevel methods II. *SIAM J. Numer. Analys.*, vol. 27, 1989, pp. 1569–1590.

[4] Babuška, I.: The finite element method with Lagrangian multipliers. *Numer. Math.*, vol. 20, 1973, pp. 179–192.

[5] Bercovier, M.: Perturbation of mixed variational problems, applications to mixed finite element methods. *R.A.I.R.O. Anal. Numer.*, vol. 12, 1978, pp. 211–236.

[6] Bramble, J.H.; Pasciak, J.E.; Xu, J.: The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms. *Math. Comp*, vol. 56, 1991, pp. 1–34.

[7] Brezzi, F.: On the existence, uniqueness and approximation of saddle point problems arising from Lagrangian multipliers. *R.A.I.R.O. Anal. Numer.*, vol. 2, 1974, pp. 129–151.

[8] Cai, Z.; Goldstein, C.I.; Pasciak, J.E.: Multilevel iteration for mixed finite element systems with penalty. *SIAM J. Sci. Stat. Comput.*, vol. 14, 1993, pp. 1072–1088.

[9] Ewing, R.E; Lazarov, R.D.; Vassilevski P.S.: Mixed finite element solutions of second order elliptic problems on grids with regular local refinement. In *Proceedings of Domain Decomposition Conference*, Moscow, 1990, pp. 206–212, SIAM, Philadelphia, 1991.

[10] Lai, C.Y.: *Analysis and Implementation of Certain Multilevel and Domain Decomposition Methods for Elliptic Problems.* Ph.D. Thesis. University of Southern California, California, 1994.

[11] Mathew, T.P.: *Domain Decomposition and Iterative Refinement Methods for Mixed Finite Element Discretisations of Elliptic Problems.* Ph.D. Thesis. Courant Institute, New York, 1989.

[12] Raviart, P.A.; Thomas, J.M.: The mixed finite element method for second order elliptic problems. In *Aspects of the Finite Element Methods.* Springer-Verlag, Heidelberg, 1977. Also in Lecture Notes in Mathematics, volume 606.

[13] Vassilevski, P.S.: Hybrid v-cycle algebraic multilevel preconditioners. *Math. Comp.*, vol. 58, 1992, pp. 489–512.

[14] Vassilevski, P.S.: Optimal order multilevel domain decomposition preconditioners. Report #1990–32, EORI, University of Wyoming, Laramie, Wyoming, 1990.

Page intentionally left blank

A CONFORMING MULTIGRID METHOD FOR THE PURE TRACTION PROBLEM OF LINEAR ELASTICITY: MIXED FORMULATION*

Chang-Ock Lee[†]
Department of Mathematics
University of Wisconsin-Madison

SUMMARY

A multigrid method using conforming P-1 finite element is developed for the two-dimensional pure traction boundary value problem of linear elasticity. The convergence is uniform even as the material becomes nearly incompressible. A heuristic argument for acceleration of the multigrid method is discussed as well. Numerical results with and without this acceleration as well as performance estimates on a parallel computer are included.

1. INTRODUCTION

Let Ω be a bounded convex polygonal domain in $\mathbf{R}^2$ and $\partial\Omega = \bigcup_{i=1}^{n} \Gamma_i$. The pure traction boundary value problem for planar linear elasticity is given in the form

$$\begin{aligned} -\underset{\sim}{\operatorname{div}}\left\{2\mu\,\underset{\approx}{\epsilon}(\underset{\sim}{u}) + \lambda\,\operatorname{tr}\left(\underset{\approx}{\epsilon}(\underset{\sim}{u})\right)\underset{\approx}{\delta}\right\} &= \underset{\sim}{f} \quad \text{in } \Omega\,, &(1)\\ \left(2\mu\,\underset{\approx}{\epsilon}(\underset{\sim}{u}) + \lambda\,\operatorname{tr}\left(\underset{\approx}{\epsilon}(\underset{\sim}{u})\right)\underset{\approx}{\delta}\right)\underset{\sim}{\nu}_i|_{\Gamma_i} &= \underset{\sim}{g}_i, \quad 1 \le i \le n\,, &(2)\end{aligned}$$

where $\underset{\sim}{u}$ denotes the displacement, $\underset{\sim}{f}$ the body force, $\underset{\sim}{g}_i$ the boundary traction, $\mu > 0$, $\lambda > 0$ the Lamé constants, and $\underset{\sim}{\nu}$ is the unit outer normal. In addition, the Lamé constants (μ, λ) belong to the range $[\mu_1, \mu_2] \times [\lambda_0, \infty)$, where μ_1, μ_2, λ_0 are fixed positive constants. The explanation for the notations used in (1) and (2) is given in [4, 6].

It is well-known that finite element method using conforming piecewise linear (P-1) finite elements converges for moderate fixed λ, and as $\lambda \to \infty$, i.e., the elastic material becomes incompressible, it seems not to converge at all ([1, 10]). In order to overcome this so called locking phenomenon, the method of reduced integration was employed by Brenner [4], Falk

*This research was partially supported by the National Science Foundation under Grant No. CDA-9024618.

[†] Current address: Department of Mathematics, Inha University, Inchon, Korea

[6] and Lee [7] in the construction of finite element methods. The finite element methods employed by them are robust in λ, i.e., they give a uniform convergence rate as $\lambda \to \infty$. In [4], Brenner proved the convergence of the P-1 nonconforming finite element method for the mixed formulation and robustness in λ using a modification of the space used by Falk in [6]. In [7], Lee proved the convergence of the P-1 conforming finite element method for the mixed formulation and robustness in λ using the same modification of the finite element space as Brenner used in [4]. In addition, Brenner adopted the W-cycle full multigrid method as a numerical solver for the resulting linear system and obtained the convergence of a multigrid method, which is robust in λ. For mixed problems without penalty term (e.g. Stokes equation), a W-cycle multigrid algorithm was developed by Verfürth [9].

In this paper we present 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 traction boundary value problem developed in [7]. We show that the convergence is uniform with respect to λ by following the argument adopted by Brenner in [4]. While the conforming multigrid method has the same order of convergence as the nonconforming multigrid method in [4], the former has about one third of the unknowns for the same mesh size. Moreover in the case of parallel computation the intergrid transfer operator of the conforming multigrid method is easier to design and has smaller communication overhead than the nonconforming one. Therefore, the conforming multigrid method promises better performance in the cases of both sequential and parallel computations. In addition, we may use this conforming multigrid method as the coarser grid correction in the multigrid algorithm for the P-1 nonconforming discretization. It gives the same convergence rate and robustness as the conforming multigrid method. In practice, V-cycle multigrid methods employing one smoothing step are convergent. Even though the P-1 conforming multigrid method is robust with respect to λ, the convergence is slow in the practical sense. Investigating the relation between eigenvalues and norms of corresponding normalized eigenfunctions $(\underset{\sim}{u}, p)$ we have found that an unusual bimodal distribution of $\| \underset{\sim}{u} \|_{\underset{\sim}{H}^1}$ vs. the eigenvalues. Based on this insight, we present a heuristic argument for a faster multigrid algorithm employing a weighting factor and a damping factor. Experimental results indicate the effectiveness of these two factors.

This paper is organized as follows. In Section 2 we explain the conforming finite element method we employ. Conforming W-cycle multigrid method is discussed in Section 3. In Section 4 we give the numerical experiments for V-cycle multigrid methods on CM-5. Also we give the performance estimate on a parallel computer. In the last section we discuss about the acceleration of multigrid algorithm and give numerical results.

2. THE FINITE ELEMENT METHOD

Throughout this paper, the letter C denotes a positive constant independent of the Lamé constants and the mesh parameter h_k, which may vary from occurrence to occurrence even in the proof of the same theorem. For the notations of several standard differential operators, refer to [4, 6].

In order for a solution of (1) and (2) to exist, $\underset{\sim}{f}$ and $\underset{\sim}{g}_i$ must satisfy the compatibility condition

$$\int_\Omega \underset{\sim}{f} \cdot \underset{\sim}{v} \; dxdy + \sum_{i=1}^{n} \int_{\Gamma_i} \underset{\sim}{g}_i \cdot \underset{\sim}{v} \; ds = 0 \quad \forall \underset{\sim}{v} \in \underset{\sim}{\mathrm{RM}}\,, \tag{3}$$

where $\underset{\sim}{\mathrm{RM}}$, the space of rigid motions, is defined by

$$\underset{\sim}{\mathrm{RM}} := \left\{ \underset{\sim}{u} \;:\; \underset{\sim}{u} = (a + by, c - bx), \quad a, b, c \in \mathbf{R} \right\} .$$

When this compatibility condition holds, the pure traction boundary value problem (1) and (2) has a unique solution $\underset{\sim}{u} \in \underset{\sim}{H}^2_\perp(\Omega)$ where

$$\underset{\sim}{H}^k_\perp(\Omega) := \left\{ \underset{\sim}{u} \in \underset{\sim}{H}^k(\Omega) \;:\; \int_\Omega \underset{\sim}{u} \cdot \underset{\sim}{v} \; dxdy = 0 \quad \forall \underset{\sim}{v} \in \underset{\sim}{\mathrm{RM}} \right\} .$$

(See [4] or Chapter 3 of [7] for more detail.) Here, $\underset{\sim}{H}^k(\Omega)$, $k \geq 0$, denotes the usual L^2-based Sobolev spaces of vector-valued functions (See [5]).

Henceforth, taking $\gamma = \frac{\lambda}{2\mu}$ and $p = \gamma \mathrm{div}\, \underset{\sim}{u}$, we consider the mixed weak formulation for (1) and (2) as follows:
Find $(\underset{\sim}{u}, p) \in \underset{\sim}{H}^1_\perp(\Omega) \times L^2(\Omega)$ such that

$$\int_\Omega \underset{\approx}{\epsilon}(\underset{\sim}{u}) : \underset{\approx}{\epsilon}(\underset{\sim}{v}) \, dxdy + \int_\Omega p(\mathrm{div}\, \underset{\sim}{v}) \, dxdy \;=\; \frac{1}{2\mu} \left[\int_\Omega \underset{\sim}{f} \cdot \underset{\sim}{v} \; dxdy + \sum_{i=1}^{n} \int_{\Gamma_i} \underset{\sim}{g}_i \cdot \underset{\sim}{v}|_{\Gamma_i} \, ds \right] , \tag{4}$$

$$\int_\Omega (\mathrm{div}\, \underset{\sim}{u}) q \, dxdy - \frac{1}{\gamma} \int_\Omega pq \, dxdy \;=\; 0 \tag{5}$$

for all $(\underset{\sim}{v}, q) \in \underset{\sim}{H}^1_\perp(\Omega) \times L^2(\Omega)$.

Replacing p and q by $\sqrt{\omega} p$ and $\sqrt{\omega} q$ $(\omega \geq 1)$, respectively, we obtain the following formulation which is equivalent to (4) and (5):
Find $(\underset{\sim}{u}, p) \in \underset{\sim}{H}^1_\perp(\Omega) \times L^2(\Omega)$ such that

$$\mathcal{B}_\omega \Big((\underset{\sim}{u}, p), (\underset{\sim}{v}, q) \Big) = \frac{1}{2\mu} \left[\int_\Omega \underset{\sim}{f} \cdot \underset{\sim}{v} \; dxdy + \sum_{i=1}^{n} \int_{\Gamma_i} \underset{\sim}{g}_i \cdot \underset{\sim}{v}|_{\Gamma_i} \, ds \right] \tag{6}$$

for all $(\underset{\sim}{v}, q) \in \underset{\sim}{H}^1_\perp(\Omega) \times L^2(\Omega)$, where

$$\mathcal{B}_\omega \Big((\underset{\sim}{u}, p), (\underset{\sim}{v}, q) \Big) := \int_\Omega \left\{ \underset{\approx}{\epsilon}(\underset{\sim}{u}) : \underset{\approx}{\epsilon}(\underset{\sim}{v}) + \sqrt{\omega} p (\mathrm{div}\, \underset{\sim}{v}) + \sqrt{\omega} (\mathrm{div}\, \underset{\sim}{u}) q - \frac{\omega}{\gamma} pq \right\} dxdy .$$

The quantity ω is called the weighting factor. Equation (6) has a unique solution on $\underset{\sim}{H}^1_\perp(\Omega) \times L^2(\Omega)$. (See [4] for more detail.)

Let $\{\mathcal{T}^k\}$ be a family of triangulations of Ω, where $\mathcal{T}^{k+1}$ is obtained by connecting the midpoints of the edges of the triangles in $\mathcal{T}^k$. Let $h_k := \max_{T\in\mathcal{T}^k} \operatorname{diam} T$, then $h_k = 2h_{k+1}$. Now let us define the conforming finite element space for our multigrid method *CMG*.

$$\begin{aligned}
\underset{\sim}{W}_k &:= \{\underset{\sim}{u} : \underset{\sim}{u}\,|_T \text{ is linear for all } T \in \mathcal{T}^k,\ \underset{\sim}{u} \text{ is continuous on } \Omega\}\,,\\
\underset{\sim}{W}_k^{\perp} &:= \left\{\underset{\sim}{u} \in \underset{\sim}{W}_k : \int_\Omega \underset{\sim}{u}\cdot\underset{\sim}{v}\ dxdy = 0 \quad \forall\, \underset{\sim}{v} \in \underset{\sim}{\mathrm{RM}}\right\}\,.
\end{aligned}$$

To describe the mixed finite element method, we define

$$Q_k := \{q : q \in L^2(\Omega) \text{ and } q|_T \text{ is a constant for all } T \in \mathcal{T}^k\}\,.$$

For the definition of nonconforming finite element space, see [4, 6].

For each k, define the bilinear form $\mathcal{B}_{\omega,k}$ on $\underset{\sim}{H}^1(\Omega)\times L^2(\Omega)$ by

$$\mathcal{B}_{\omega,k}\left((\underset{\sim}{u},p),(\underset{\sim}{v},q)\right) := \int_\Omega \left\{\underset{\approx}{\epsilon}(\underset{\sim}{u}) : \underset{\approx}{\epsilon}(\underset{\sim}{v}) + \sqrt{\omega}p(P_{k-1}\operatorname{div}\underset{\sim}{v}) + \sqrt{\omega}(P_{k-1}\operatorname{div}\underset{\sim}{u})q - \frac{\omega}{\gamma}pq\right\} dxdy\,,$$

where P_{k-1} is the L^2-orthogonal projection onto Q_{k-1}. Now, we have a conforming discretization of (6), which are modifications of one proposed by Falk in [6]:
Find $(\underset{\sim}{u}_k, p_k) \in \underset{\sim}{W}_k^{\perp}\times Q_{k-1}$ such that

$$\mathcal{B}_{\omega,k}\left((\underset{\sim}{u}_k,p_k),(\underset{\sim}{v},q)\right) = \frac{1}{2\mu}\left[\int_\Omega \underset{\sim}{f}\cdot\underset{\sim}{v}\ dxdy + \sum_{i=1}^{n}\int_{\Gamma_i} \underset{\sim}{g}_i\cdot\underset{\sim}{v}\,|_{\Gamma_i}\,ds\right] \tag{7}$$

for all $(\underset{\sim}{v},q) \in \underset{\sim}{W}_k^{\perp}\times Q_{k-1}$.

In Chapter 3 of [7], proving the analogue of the classical lemma for the existence of an inverse of the divergence operator Lee showed the uniqueness of the solution of the conforming discretization (7) with $\omega = 1$ and derived the following discretization error estimate:

$$\begin{aligned}
\|\underset{\sim}{u}-\underset{\sim}{u}_k\|_{\underset{\sim}{L}^2(\Omega)} &+ h_k\left(|\underset{\sim}{u}-\underset{\sim}{u}_k|_{\underset{\sim}{H}^1(\Omega)} + \|p-p_k\|_{L^2(\Omega)}\right)\\
&\le Ch_k^2\left\{\|\underset{\sim}{f}\|_{\underset{\sim}{L}^2(\Omega)} + \sum_{i=1}^{n}\|\underset{\sim}{g}_i\|_{\underset{\sim}{H}^{1/2}(\Gamma_i)}\right\}\,.
\end{aligned}$$

In [4], Brenner showed the uniqueness of the solution of the nonconforming discretization and derived a similar discretization error estimate.

3. THE CONFORMING MULTIGRID ALGORITHM

In this section we present lemmas and theorems without proofs which are found in Chapter 4 of [7]. We set $\omega = 1$ for the time being until we have a statement for $\omega > 1$. Let $\mathcal{B} = \mathcal{B}_1$ and $\mathcal{B}_k = \mathcal{B}_{1,k}$.

Define the mesh dependent inner product by

$$\Big((\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big)_k := (\underset{\sim}{u},\underset{\sim}{v})_{\underset{\sim}{L}^2(\Omega)} + h_k^2(p,q)_{L^2(\Omega)}\,.$$

The intergrid transfer operator $I_k^{k-1} : \underset{\sim}{W}_k \times Q_{k-1} \to \underset{\sim}{W}_{k-1} \times Q_{k-2}$ is defined by

$$\Big(I_k^{k-1}(\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big)_{k-1} = \Big((\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big)_k$$

for all $(\underset{\sim}{u},p) \in \underset{\sim}{W}_k \times Q_{k-1}$, and $(\underset{\sim}{v},q) \in \underset{\sim}{W}_{k-1} \times Q_{k-2}$.

Lemma 1 $I_k^{k-1} : \underset{\sim}{W}_k^\perp \times Q_{k-1} \to \underset{\sim}{W}_{k-1}^\perp \times Q_{k-2}\,.$ □

Define $B_k : \underset{\sim}{W}_k \times Q_{k-1} \to \underset{\sim}{W}_k \times Q_{k-1}$ by

$$\Big(B_k(\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big)_k = \mathcal{B}_k\Big((\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big) \quad \forall(\underset{\sim}{u},p),(\underset{\sim}{v},q) \in \underset{\sim}{W}_k \times Q_{k-1}\,.$$

Lemma 2 $B_k : \underset{\sim}{W}_k \times Q_{k-1} \to \underset{\sim}{W}_k^\perp \times Q_{k-1}\,.$ □

Let $B_k^\perp = B_k|_{\underset{\sim}{W}_k^\perp \times Q_{k-1}}$.

Lemma 3 *The spectral radius of* $B_k^\perp \le Ch_k^{-2}\,.$ □

The mesh-dependent norms on $\underset{\sim}{W}_k^\perp \times Q_{k-1}$ are defined as follows:

$$|\!|\!|(\underset{\sim}{u},p)|\!|\!|_{s,k} := \sqrt{\left(\left({B_k^\perp}^2\right)^{s/2}(\underset{\sim}{u},p),(\underset{\sim}{u},p)\right)_k} \quad \forall(\underset{\sim}{u},p) \in \underset{\sim}{W}_k^\perp \times Q_{k-1}\,.$$

Define $P_k^{k-1} : \underset{\sim}{W}_k^\perp \times Q_{k-1} \to \underset{\sim}{W}_{k-1}^\perp \times Q_{k-2}$ by

$$\mathcal{B}_{k-1}\Big(P_k^{k-1}(\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big) = \mathcal{B}_k\Big((\underset{\sim}{u},p),(\underset{\sim}{v},q)\Big)$$

for all $(\underset{\sim}{u},p) \in \underset{\sim}{W}_k^\perp \times Q_{k-1}$ and $(\underset{\sim}{v},q) \in \underset{\sim}{W}_{k-1}^\perp \times Q_{k-2}$.

The k-th level iteration scheme of the conforming multigrid algorithm: The k-th level iteration with initial guess $(\underset{\sim}{y}_0, z_0) \in \underset{\sim}{W}_k^\perp \times Q_{k-1}$ yields $CMG(k,(\underset{\sim}{y}_0,z_0),(\underset{\sim}{w},r))$ as a conforming approximate solution to the following problem.

Find $(\underset{\sim}{y}, z) \in \underset{\sim}{W}_k^{\perp} \times Q_{k-1}$ such that

$$B_k^{\perp}(\underset{\sim}{y}, z) = (\underset{\sim}{w}, r), \text{ where } (\underset{\sim}{w}, r) \in \underset{\sim}{W}_k^{\perp} \times Q_{k-1}\,.$$

For $k = 1$, $CMG(1, (\underset{\sim}{y}_0, z_0), (\underset{\sim}{w}, r))$ is the solution obtained from a direct method, i.e.,

$$CMG(1, (\underset{\sim}{y}_0, z_0), (\underset{\sim}{w}, r)) = \left(B_1^{\perp}\right)^{-1} (\underset{\sim}{w}, r)\,.$$

For $k > 1$,

Smoothing Step: the approximation $(\underset{\sim}{y}_m, z_m) \in \underset{\sim}{W}_k^{\perp} \times Q_{k-1}$ is constructed recursively from the initial guess $(\underset{\sim}{y}_0, z_0)$ and the equations

$$(\underset{\sim}{y}_l, z_l) = (\underset{\sim}{y}_{l-1}, z_{l-1}) + \frac{1}{\Lambda_k^2} B_k((\underset{\sim}{w}, r) - B_k(\underset{\sim}{y}_{l-1}, z_{l-1})), \quad 1 \le l \le m\,.$$

Here, $\Lambda_k := Ch_k^{-2}$ is greater than or equal to the spectral radius of $B_k^{\perp}$, and m is an integer to be determined later.

Correction Step: The coarser-grid correction in $\underset{\sim}{W}_{k-1}^{\perp} \times Q_{k-2}$ is obtained by applying the $(k-1)$-th level conforming iteration twice. In other words, it is the standard W-cycle multigrid method with $\mu = 2$. More precisely,

$$\begin{aligned} (\underset{\sim}{v}_0, q_0) &= (\underset{\sim}{0}, 0) \quad \text{and} \\ (\underset{\sim}{v}_i, q_i) &= CMG(k-1, (\underset{\sim}{v}_{i-1}, q_{i-1}), (\bar{\underset{\sim}{w}}, \bar{r})), \quad i = 1, 2 \end{aligned}$$

where $(\bar{\underset{\sim}{w}}, \bar{r}) \in W_{k-1}^{\perp} \times Q_{k-2}$ is defined by $(\bar{\underset{\sim}{w}}, \bar{r}) := I_k^{k-1}\left((\underset{\sim}{w}, r) - B_k(\underset{\sim}{y}_m, z_m)\right)$.

Then

$$CMG(k, (\underset{\sim}{y}_0, z_0), (\underset{\sim}{w}, r)) = (\underset{\sim}{y}_m, z_m) + (\underset{\sim}{v}_2, q_2)\,.$$

Let the final output of the two-grid algorithm be

$$(\underset{\sim}{y}^{\#}, z^{\#}) := (\underset{\sim}{y}_m, z_m) + (\underset{\sim}{v}^{\#}, q^{\#})$$

where

$$(\underset{\sim}{v}^{\#}, q^{\#}) = \left(B_{k-1}^{\perp}\right)^{-1} I_k^{k-1} B_k(\underset{\sim}{y} - \underset{\sim}{y}_m, z - z_m)\,.$$

Lemma 4 $(\underset{\sim}{v}^{\#}, q^{\#}) = P_k^{k-1}(\underset{\sim}{y} - \underset{\sim}{y}_m, z - z_m)\,.$ □

Let

$$R_k := I - \frac{1}{\Lambda_k^2}\left(B_k\right)^2 .$$

Then we have

$$\begin{aligned}(\underset{\sim}{y} - \underset{\sim}{y}_m, z - z_m) &= R_k^m(\underset{\sim}{y} - \underset{\sim}{y}_0, z - z_0), \\ (\underset{\sim}{y} - \underset{\sim}{y}^{\#}, z - z^{\#}) &= (I - P_k^{k-1})R_k^m(\underset{\sim}{y} - \underset{\sim}{y}_0, z - z_0) .\end{aligned}$$

Lemma 5 (Smoothing Step) *There exists a constant C, independent of h_k and m, such that*

$$|||R_k^m(\underset{\sim}{u}, p)|||_{2,k} \le C h_k^{-2}\frac{1}{\sqrt{m}}|||(\underset{\sim}{u}, p)|||_{0,k} \quad \forall(\underset{\sim}{u}, p) \in \underset{\sim}{W}_k^{\perp} \times Q_{k-1} . \quad \Box$$

Lemma 6 (Approximation Step) *There exists a constant C, independent of h_k and m, such that*

$$|||(I - P_k^{k-1})(\underset{\sim}{u}, p)|||_{0,k} \le C h_k^2 |||(\underset{\sim}{u}, p)|||_{2,k} \quad \forall(\underset{\sim}{u}, p) \in \underset{\sim}{W}_k^{\perp} \times Q_{k-1} . \quad \Box$$

Theorem 1 (Convergence of the Two-Grid Algorithm) *There exists a constant C, independent of k and m, such that*

$$|||(\underset{\sim}{y} - \underset{\sim}{y}^{\#}, z - z^{\#})|||_{0,k} \le \frac{C}{\sqrt{m}}|||(\underset{\sim}{y} - \underset{\sim}{y}_0, z - z_0)|||_{0,k} . \quad \Box$$

Theorem 2 (Convergence of the k-th Level Iteration) *There exists a constant C, independent of k and m, such that*

$$|||(\underset{\sim}{y}, z) - CMG(k, (\underset{\sim}{y}_0, z_0), (\underset{\sim}{w}, r))|||_{0,k} \le \frac{C}{\sqrt{m}}|||(\underset{\sim}{y} - \underset{\sim}{y}_0, z - z_0)|||_{0,k} . \quad \Box$$

4. EXPERIMENTAL RESULTS

For our numerical experiments, we choose the model problem studied in [4]:

$$\begin{aligned} -\underset{\sim}{\mathrm{div}}\left\{\underset{\approx}{\epsilon}(\underset{\sim}{u}) + \lambda \operatorname{tr}\left(\underset{\approx}{\epsilon}(\underset{\sim}{u})\right)\underset{\approx}{\delta}\right\} &= \underset{\sim}{f} \quad \text{in } \Omega = \text{unit square}, \\ \left(\underset{\approx}{\epsilon}(\underset{\sim}{u}) + \lambda \operatorname{tr}\left(\underset{\approx}{\epsilon}(\underset{\sim}{u})\right)\underset{\approx}{\delta}\right)\underset{\sim}{\nu}_i|_{\Gamma_i} &= \underset{\sim}{g}_i, \quad 1 \le i \le 4 ,\end{aligned}$$

where Γ_i $(1 \leq i \leq 4)$ represents four sides of the unit square. The body force $\underset{\sim}{f} = (f_1, f_2)^t$ is defined by

$$\begin{aligned} f_1 &= -\pi^2 \sin \pi x \sin \pi y + 2\pi^2 \left(\frac{1}{\lambda} + 1\right) \cos \pi x \sin \pi y \,, \\ f_2 &= -\pi^2 \cos \pi x \cos \pi y + 2\pi^2 \left(\frac{1}{\lambda} + 1\right) \sin \pi x \cos \pi y \end{aligned}$$

and the boundary tractions are defined by

$$\begin{aligned} \underset{\sim}{g}_1 &= \left(-\frac{\pi}{\lambda} \cos \pi x, 0\right)^t, & \underset{\sim}{g}_2 &= \left(\pi \sin \pi y, -\frac{\pi}{\lambda} \cos \pi y\right)^t, \\ \underset{\sim}{g}_3 &= \left(-\frac{\pi}{\lambda} \cos \pi x, 0\right)^t, & \underset{\sim}{g}_4 &= \left(\pi \sin \pi y, -\frac{\pi}{\lambda} \cos \pi y\right)^t. \end{aligned}$$

The exact solution $\underset{\sim}{u} = (u_1, u_2)^t \in \underset{\sim}{H}^2_\perp(\Omega)$ is

$$\begin{aligned} u_1 &= \left(-\sin \pi x + \frac{1}{\lambda} \cos \pi x\right) \sin \pi y + \frac{4}{\pi^2} \,, \\ u_2 &= \left(-\cos \pi x + \frac{1}{\lambda} \sin \pi x\right) \cos \pi y \,. \end{aligned}$$

First, we describe the implementation of conforming multigrid method *CMG*. Let ϕ_i^k be the piecewise linear function which equals 1 at exactly one vertex p_i and equals 0 at all other vertices of $T \in \mathcal{T}_k$ and ψ_i^k be the piecewise constant function which equals 1 on exactly one triangle T_i and equals 0 on all other triangles of $\mathcal{T}_k$. Then

$$\{\underset{\sim}{\Phi}_i^k\}_{1 \leq i \leq n_k} = \{(\phi_i^k, 0, 0), (0, \phi_j^k, 0), (0, 0, \psi_l^{k-1})\}$$

forms a conforming basis of $\underset{\sim}{W}_k \times Q_{k-1}$. The matrix representation of B_k with respect to the basis $\{\underset{\sim}{\Phi}_i^k\}_{1 \leq i \leq n_k}$ in the *CMG* algorithm is equal to $M_k^{-1} S_k$ where M_k is the mass matrix and S_k is the stiffness matrix. Let E_k^{k-1} be the matrix representation of the intergrid transfer operator I_k^{k-1}. Then we have

$$E_k^{k-1} = M_{k-1}^{-1} (E_{k-1}^k)^t M_k$$

where E_{k-1}^k is the matrix representation of the natural imbedding from $\underset{\sim}{W}_{k-1}^\perp \times Q_{k-2}$ into $\underset{\sim}{W}_k^\perp \times Q_{k-1}$. Let X_k be the vector space which consists of the coefficients of the functions in $\underset{\sim}{W}_k \times Q_{k-1}$ with respect to the basis $\{\underset{\sim}{\Phi}_i^k\}_{1 \leq i \leq n_k}$. Similarly we define $X_k^\perp$ as the equivalent vector space to $\underset{\sim}{W}_k^\perp \times Q_{k-1}$. With the compatibility condition (3), the *CMG* algorithm can be rewritten in matrix form for the following problem:

Find $(\underset{\sim}{Y}, Z) \in X_k^\perp$ such that

$$(M_k^{-1} S_k)|_{X_k^\perp} (\underset{\sim}{Y}, Z)^t = (\underset{\sim}{W}, R)^t, \quad \text{where} \quad (\underset{\sim}{W}, R)^t \in X_k^\perp \,.$$

For $k = 1$, $CMG(1, (\underset{\sim}{Y}_0, Z_0), (\underset{\sim}{W}, R))$ is the solution obtained from a direct method, i.e.,

$$CMG(1, (\underset{\sim}{Y}_0, Z_0), (\underset{\sim}{W}, R)) = (M_1^{-1} S_1)|_{X_1^\perp}^{-1} (\underset{\sim}{W}, R)\,.$$

For $k > 1$,

Smoothing Step: the approximation $(\underset{\sim}{Y}_m, Z_m) \in X_k^\perp$ is constructed recursively from the initial guess $(\underset{\sim}{Y}_0, Z_0) \in X_k^\perp$ and the equations

$$(\underset{\sim}{Y}_l, Z_l) = (\underset{\sim}{Y}_{l-1}, Z_{l-1}) + \frac{1}{\Lambda_k^2} M_k^{-1} S_k((\underset{\sim}{W}, R) - M_k^{-1} S_k(\underset{\sim}{Y}_{l-1}, Z_{l-1})), \quad 1 \le l \le m\,.$$

Here, $\Lambda_k := C h_k^{-2}$ is greater than or equal to the spectral radius of $(M_k^{-1} S_k)|_{X_k^\perp}$, and m is an integer to be determined later.

Correction Step: The coarser-grid correction in $X_{k-1}^\perp$ is obtained by applying the $(k-1)$-th level conforming iteration twice. In other words, it is the standard W-cycle multigrid method. More precisely,

$$\begin{aligned} (\underset{\sim}{V}_0, Q_0) &= (\underset{\sim}{0}, 0) \quad \text{and} \\ (\underset{\sim}{V}_i, Q_i) &= CMG(k-1, (\underset{\sim}{V}_{i-1}, Q_{i-1}), (\bar{\underset{\sim}{W}}, \bar{R})), \quad i = 1, 2 \end{aligned}$$

where $(\bar{\underset{\sim}{W}}, \bar{R}) \in X_{k-1}^\perp$ is defined by

$$(\bar{\underset{\sim}{W}}, \bar{R}) := E_k^{k-1} \left((\underset{\sim}{W}, R) - M_k^{-1} S_k (\underset{\sim}{Y}_m, Z_m) \right)\,.$$

Then

$$CMG(k, (\underset{\sim}{Y}_0, Z_0), (\underset{\sim}{W}, R)) = (\underset{\sim}{Y}_m, Z_m) + E_{k-1}^k (\underset{\sim}{V}_2, Q_2)\,.$$

With respect to the basis $\{\underset{\sim}{\Phi}_i^k\}_{1 \le i \le n_k}$ the mass matrix M_k has seven entries per row so that it is costly to take inverse of M_k in the implementation of the algorithm at each level of the multigrid. In practice, we replace M_k by an appropriate N_k satisfying

(i) M_k and N_k are spectrally equivalent, i.e., there is a constant β, independent of h_k, such that

$$0 < \beta^{-1} \le \frac{(N_k \underset{\sim}{U}, \underset{\sim}{U})_{l_2}}{(M_k \underset{\sim}{U}, \underset{\sim}{U})_{l_2}} \le \beta \quad \forall \underset{\sim}{U} \in X_k,\ \underset{\sim}{U} \ne \underset{\sim}{0}\,.$$

(ii)

$$N_k^{-1} S_k : X_k \to X_k^\perp\,.$$

(iii)

$$N_{k-1}^{-1} (E_{k-1}^k)^t N_k : X_k \to X_{k-1}^\perp\,.$$

The conditions (ii) and (iii) are essential because the solution of our problem should lie in $X_k^\perp$. In the smoothing step, instead of Λ_k, we use $\Lambda_{N_k,k}$ which is the spectral radius of $(N_k^{-1}S_k)|_{X_k^\perp}$ and by Lemma 3 we have

$$\text{Spectral Radius of } (N_k^{-1}S_k)|_{X_k^\perp} \leq Ch_k^{-2} .$$

The multigrid algorithm *CMG* is convergent with respect to the norm

$$|\!|\!| \cdot |\!|\!|_{0,k} = \sqrt{(\cdot,\cdot)_k} = \sqrt{(M_k\cdot,\cdot)_{l_2}} .$$

By slight modification of the proof of the convergence theorem of the *CMG* algorithm, we obtain the convergence theorem of the multigrid algorithm containing N_k instead of M_k with respect to $(N_k\cdot,\cdot)_{l_2}^{1/2}$ which is equivalent to $|\!|\!| \cdot |\!|\!|_{0,k}$. See [2] for more detail. For this specific experiment on the unit square we take $N_k = \text{diag}(M_k)$ as suggested in [2], which allows the use of an under-relaxed Jacobi scheme of smoothing. Most rows of the stiffness matrix S_k have sixteen entries so that most rows of $N_k^{-1}S_k$ have again sixteen entries. Note that the matrix representation for I_k^{k-1} has again seven entries per row. In the coarsest grid we use a direct solver for the (6×6) linear system which comes from the matrix representation $B_1^\perp$ with respect to the basis of $X_1^\perp$.

The performance of multigrid algorithms has usually been measured in *Work Units.* In serial machines, since the total CPU time is proportional to the amount of computational work and smoothing steps make up most of the multigrid work, a reasonable unit of effort is the *Work Unit* (WU) defined in [3] as the amount of computations in one smoothing step in the finest grid.

However, in parallel machines (in particular, massively parallel machines adopting data parallelism) we use a somewhat different method to measure the computational work. In this paper, we use one WU as the amount of computations needed in one smoothing step of the *conforming* multigrid method *CMG* at the *finest grid* on a *serial machine.* Let n_k be the number of unknowns at k-th level and Q_{comp} be the number of operations required to compute one smoothing step at each mesh point. Then we have

$$n_J Q_{comp} = 1 \quad \text{(WU)}$$

where J-th level represents the finest grid. Let p be the number of processors and assume two-dimensional square data distribution (cf. Chapter 5 of [7]). Then the number of unknowns of k-th level allocated to each processor is

$$r_k = \left\lceil \frac{n_k}{p} \right\rceil , \quad \text{and} \quad n_k = \left(\frac{1}{4}\right)^{J-k} \cdot n_J \quad \text{for} \quad k = 1, \ldots, J ,$$

where $\lceil x \rceil$ is the smallest integer greater than x. On a parallel machine we need an additional unit to measure the communication work. We define one CU (*Communication Unit*) as the amount of communications needed in one smoothing step of the *conforming* multigrid method *CMG* when we assume a large system of $p \cdot n_J$ number of unknowns. Let Q_{comm} be the number of interprocessor communication steps required to compute one smoothing step at each mesh point. Since about $4\sqrt{r_k}$ mesh points in a processor do interprocessor communication in the

Table I: V-cycle of *CMG* when $h = 1/64$

	$\lambda = 10$			$\lambda = 100$			$\lambda = 1000$		
smoothing	WU	CU	iter	WU	CU	iter	WU	CU	iter
1	68	582	1626	244	2073	5788	334	2842	7935
2	67	572	798	223	1894	2645	293	2491	3478
3	66	559	520	201	1714	1595	255	2169	2019
4	64	546	381	184	1564	1092	226	1924	1343

smoothing step of the conforming multigrid method *CMG*, we have

$$4\sqrt{n_J}Q_{comm} = 1 \quad (\mathrm{CU}).$$

Let T_{comp} be the time needed to perform the computational work of one smoothing step at one mesh point and T_{comm} be the time needed to perform the interprocessor communication in one smoothing step at one mesh point. The multigrid algorithms in this paper are one-sided method, i.e., it uses the smoothing step before correction step. If smoothing steps are used before and after correction step, the multigrid method is called symmetric. Note that as far as the convergence is concerned a symmetric V-cycle multigrid iteration is the same as two one-sided V-cycle iterations (See [8]).

The programs execute the multigrid iterations until the discrete L_2 relative error is less than .03 for the mesh size $h = \frac{1}{64}$ (10,498 unknowns) and for various number of smoothings and λ. The experiments reported here were run in double-precision arithmetic on CM-5 Vector Units with 32 processors.

In the Table I, the numbers in the columns of $\lambda = 10, 100, 1000$ represent Work Units, Communication Units and N_{iter} (the number of iterations of *CMG*). While we have only proven that *CMG* converges for the W-cycle with many smoothing steps, we see that in practice it converges even for the V-cycle with one smoothing step. In both cases, convergence is independent of the mesh size h_k and Lamé constant λ. The total amount of computational work of a 7-level V-cycle is

$$\mathcal{W}_{comp} = m\left(\sum_{k=2}^{7}\left\lceil\frac{\left(\frac{1}{4}\right)^{7-k} n_7}{p}\right\rceil\right)\frac{N_{iter}}{n_7}.$$

The total amount of communication of the 7-level V-cycle is

$$\mathcal{W}_{comm} = m\left(\sum_{k=2}^{7}\sqrt{\left\lceil\frac{\left(\frac{1}{4}\right)^{7-k} n_7}{p}\right\rceil}\right)\frac{N_{iter}}{\sqrt{n_7}}.$$

The total elapsed time is

$$T = \mathcal{W}_{comp} T_{comp} + \mathcal{W}_{comm} T_{comm} \, .$$

Therefore the performance of the multigrid algorithm is dependent upon the ratio between T_{comp} and T_{comm}. It is not easy to obtain the ratio because it heavily depends on the implementation of algorithms, e.g., the topology of data distribution and distance of communications.

5. ACCELERATION OF MULTIGRID METHOD

Even though the P-1 conforming multigrid method is robust with respect to λ, the convergence is slow in any practical sense. In this section we present a heuristic argument for the acceleration of the multigrid algorithm *CMG*.

Replacing p and q by $\sqrt{\omega}p$ and $\sqrt{\omega}q$ ($\omega > 1$), respectively, we use the argument in Chapter 3 of [7] to show the uniqueness of the solution of the equations (6) and (7), and to derive the following discretization error estimate:

$$\| \underset{\sim}{u} - \underset{\sim}{u}_k \|_{\underset{\sim}{L}^2(\Omega)} + h_k \left(| \underset{\sim}{u} - \underset{\sim}{u}_k |_{\underset{\sim}{H}^1(\Omega)} + \sqrt{\omega} \| p - p_k \|_{L^2(\Omega)} \right)$$
$$\leq C_\omega h_k^2 \left\{ \| \underset{\sim}{f} \|_{\underset{\sim}{L}^2(\Omega)} + \sum_{i=1}^{n} \| \underset{\sim}{g}_i \|_{\underset{\sim}{H}^{1/2}(\Gamma_i)} \right\} .$$

Also, following the argument in Section 3, we can develop the same multigrid algorithm for the problem:
Find $(\underset{\sim}{y}, z) \in \underset{\sim}{W}_k^\perp \times Q_{k-1}$ such that

$$B_{\omega,k}^\perp(\underset{\sim}{y}, z) = (\underset{\sim}{w}, r), \text{ where } (\underset{\sim}{w}, r) \in \underset{\sim}{W}_k^\perp \times Q_{k-1} \, .$$

For positive definite systems of which energy norms are equivalent to H^1 norm, the normalized eigenfunctions (with respect to L^2 norm) corresponding to the large eigenvalues have large H^1 norm, which means that these eigenfunctions are highly oscillatory. However our linear system induced from the mixed finite element discretization of the pure traction problem is indefinite. Moreover, the solution space is composed of two different spaces. One is the space of piecewise linear functions and another is the space of piecewise constant functions. Using MATLAB we have investigated the relation between eigenvalues and $\| \underset{\sim}{u} \|_{\underset{\sim}{H}^1}$ and $\|[p]\|_{L^2}$ of normalized eigenfunctions $(\underset{\sim}{u}, p)$ (with respect to $\| \cdot \|_{0,k}$) where $[p]$ represents the jump across the edges of each triangle in $\mathcal{T}_{k-1}$. Figure 1 shows the eigenvalues and $\| \underset{\sim}{u} \|_{\underset{\sim}{H}^1}$ and $\|[p]\|_{L^2}$ of normalized eigenfunctions of $N_k^{-1} S_k$ where $h = 1/16$ (706 unknowns). The eigenvectors corresponding to the negative eigenvalues have large $\|[p]\|_{L^2}$, which means p is highly oscillating, so that the error of p corresponding to the negative eigenvalues is not reduced by smoothing step enough to be corrected in the correction step. By introducing the

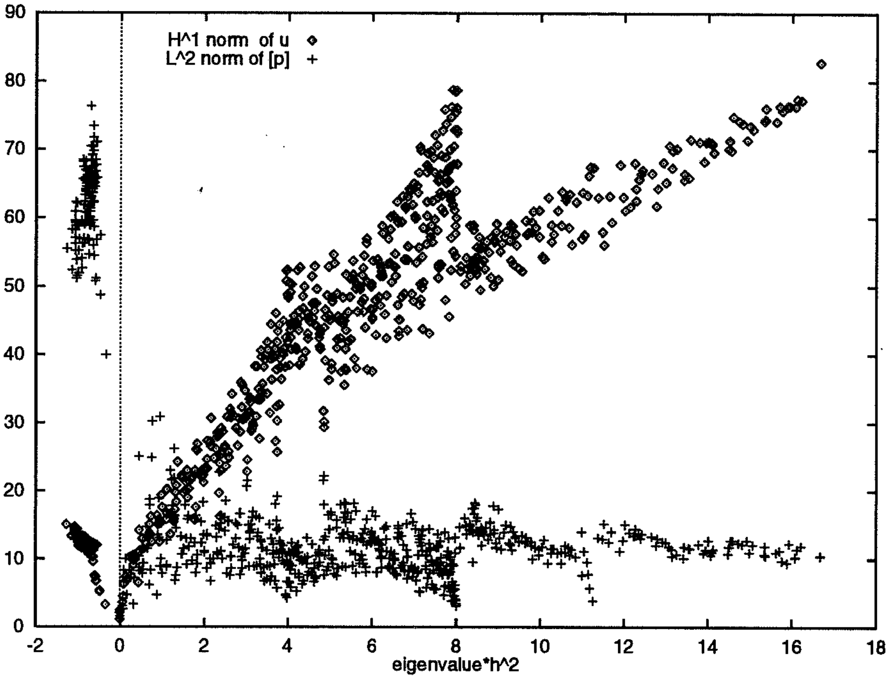


Figure 1: $h = \frac{1}{16}$, $\lambda = 1000, \omega = 1$

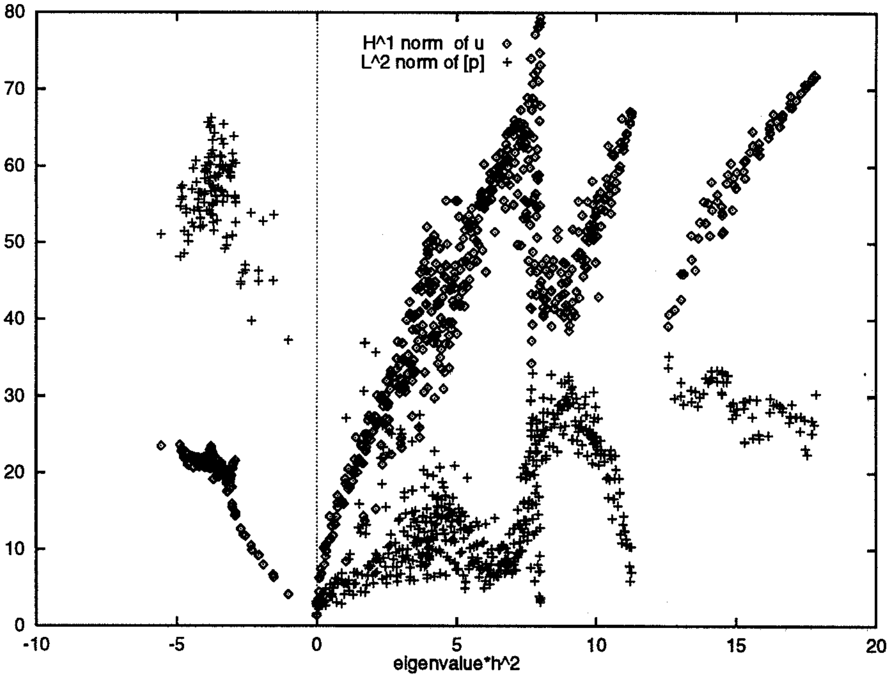


Figure 2: $h = \frac{1}{16}$, $\lambda = 1000, \omega = 7$

weighting factor, we can magnify the size of the negative eigenvalues with little effect on the general distribution of eigenvalues. Figure 2 shows the eigenvalues and $\| \underset{\sim}{u} \|_{\underset{\sim}{H}^1}$ and $\|[p_\omega]\|_{L^2}$ of normalized eigenfunctions of $N_k^{-1} S_{\omega,k}$ with weighting $\omega = 7$. By employing such a weighting factor the magnitudes of negative eigenvalues become larger while that of positive eigenvalues grow little. Therefore we expect the better performance of multigrid method for the system with the weighting factor.

Since we use the Gershgorin theorem to estimate the maximum eigenvalue of $N_k^{-1} S_{\omega,k}$, we always over-estimate it. Therefore for acceleration of our multigrid algorithm, it is useful to use damping factor θ in the smoothing step as follows:

$$(\underset{\sim}{y}_l, z_l) = (\underset{\sim}{y}_{l-1}, z_{l-1}) + \frac{\theta^2}{\Lambda_k^2} B_{\omega,k} \left((\underset{\sim}{w}, r) - B_{\omega,k}(\underset{\sim}{y}_{l-1}, z_{l-1}) \right), \quad 1 \leq l \leq m .$$

There is one more reason why the damping factor is useful. In Figures 1 and 2, there are two or three peaks of $\| \underset{\sim}{u} \|_{\underset{\sim}{H}^1}$, which means that the error of $\underset{\sim}{u}$ corresponding to mid-ranged positive eigenvalues is not reduced by smoothing step enough to be corrected in the correction step. By using a damping factor the error of $\underset{\sim}{u}$ corresponding to several peaks can be reduced simultaneously in the smoothing step. Numerical results for the effect of the weighting and damping factors are shown in Tables II–IV. However, as $\theta \to 2$, the multigrid algorithm is suddenly divergent so that it is risky to take $\theta \approx 2$ in order to get better convergence results. Tables V–VII show the convergence results with 2 smoothings with $\theta = 1$ for the first smoothing and $\theta = \alpha$ for the second smoothing. By the alternating smoothings, we can take θ near 2 in safe. Using these weighting and damping factors, we get about 30 times faster results.

Acknowledgements. I would like to thank Professor S. V. Parter for his valuable advice and encouragement.

REFERENCES

1. Babuška, I.; and Szabo, B.: On the rates of convergence of the finite element method. *Internat. J. Numer. Meth. Engng.*, vol. 18, 1982, pp. 323-341.

2. Bank, R. E.; and Dupont, T.: An optimal order process for solving finite element equations. *Math. Comp.*, vol. 36, 1981, pp. 35-51.

3. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, vol. 31, 1977, pp. 333-390.

4. Brenner, S.: A nonconforming mixed multigrid method for the pure traction problem in planar linear elasticity. *Math. Comp.*, vol. 63, 1994, pp. 435-460 & S1-S5.

5. Ciarlet, P.: *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978.

6. Falk, R. S.: Nonconforming finite element methods for the equations of linear elasticity. *Math. Comp.*, vol. 57, 1991, pp. 529-550.

Table II: V-cycle of *CMG* with one smoothing, $\lambda = 10$ and $h = 1/64$

	$\omega = 1$			$\omega = 3$			$\omega = 4$			$\omega = 10$		
θ	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	68	582	1626	13	111	310	13	112	313	19	160	446
1.2	49	419	1171	9	78	217	9	78	217	13	111	309
1.4	38	325	907	7	58	161	7	57	160	10	81	226
1.6	32	271	758	5	45	126	5	44	122	7	62	173
1.8	divergent			4	37	103	4	35	97	6	49	136
2.0	divergent			divergent			3	29	81	5	39	110

Table III: V-cycle of *CMG* with one smoothing, $\lambda = 100$ and $h = 1/64$

	$\omega = 1$			$\omega = 6$			$\omega = 7$			$\omega = 10$		
θ	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	244	2073	5788	16	136	380	16	139	387	18	157	439
1.2	210	1788	4992	11	96	268	11	97	270	13	109	304
1.4	353	3001	8381	9	73	203	8	72	200	9	80	223
1.6	divergent			7	61	169	7	57	159	7	61	170
1.8	divergent			11	90	252	6	53	147	6	49	136
2.0	divergent			divergent			divergent			divergent		

Table IV: V-cycle of *CMG* with one smoothing, $\lambda = 1000$ and $h = 1/64$

	$\omega = 1$			$\omega = 7$			$\omega = 8$			$\omega = 10$		
θ	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	334	2841	7935	17	144	401	17	146	408	19	158	440
1.2	336	2855	7972	12	101	283	12	102	285	13	109	305
1.4	divergent			9	78	217	9	76	213	9	80	224
1.6	divergent			8	67	187	7	62	173	7	62	173
1.8	divergent			divergent			10	88	247	6	53	147
2.0	divergent			divergent			divergent			divergent		

Table V: V-cycle of *CMG* with alternating smoothings, $\lambda = 10$ and $h = 1/64$

	$\omega = 1$			$\omega = 3$			$\omega = 4$			$\omega = 10$		
α	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	67	572	798	13	112	156	13	113	158	19	160	224
1.2	56	473	661	11	92	128	11	92	129	15	131	183
1.4	47	397	555	9	76	106	9	77	107	13	108	151
1.6	40	340	475	7	64	89	7	64	89	11	90	126
1.8	35	298	416	6	54	76	6	54	75	9	76	106
2.0	32	271	378	6	47	66	5	46	64	8	64	90

Table VI: V-cycle of *CMG* with alternating smoothings, $\lambda = 100$ and $h = 1/64$

	$\omega = 1$			$\omega = 6$			$\omega = 7$			$\omega = 10$		
α	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	223	1894	2645	16	136	190	16	138	193	18	158	220
1.2	191	1620	2263	13	112	156	13	114	159	15	129	180
1.4	172	1461	2040	11	93	130	11	94	131	12	106	148
1.6	170	1445	2017	9	79	110	9	79	110	10	88	123
1.8	232	1977	2761	8	69	96	8	67	94	9	74	103
2.0	divergent			8	64	90	7	59	83	7	63	88

Table VII: V-cycle of *CMG* with alternating smoothings, $\lambda = 1000$ and $h = 1/64$

	$\omega = 1$			$\omega = 7$			$\omega = 8$			$\omega = 10$		
α	WU	CU	iter	WU	CU	iter	WU	CU	iter	WU	CU	iter
1.0	293	2491	3478	17	143	199	17	146	204	19	158	220
1.2	255	2171	3032	14	117	164	14	120	167	15	129	180
1.4	241	2049	2861	12	98	137	12	100	139	12	106	148
1.6	271	2308	3223	10	83	116	10	83	116	10	89	124
1.8	divergent			9	74	103	8	72	100	9	74	104
2.0	divergent			9	73	102	8	65	91	8	64	90

7. Lee, C.-O.: *Multigrid methods and parallel computations for elliptic problems: with an emphasis on linear elasticity*, Ph.D. thesis, Univ. of Wisconsin, Madison, WI, 1995.

8. McCormick, S.: Multigrid methods for variational problems: Further results. *SIAM J. Numer. Anal.*, vol. 21, 1984, pp. 255-263.

9. Verfürth, R.: A multilevel algorithm for mixed problems. *SIAM J. Numer. Anal.*, vol. 21, 1984, pp. 264-271.

10. Vogelius, M.: An analysis of the *p*-version of the finite element method for nearly incompressible materials. Uniformly valid, optimal error estimates. *Numer. Math.*, vol. 41, 1983, pp. 39-53.

Page intentionally left blank

MULTIPLE SCALE SIMULATION FOR TRANSITIONAL AND TURBULENT FLOW

Chaoqun Liu* and Zhining Liu†
Numerical Simulation Group, Department of Mathematics
University of Colorado at Denver
Denver, CO

SUMMARY

A new concept, multiple scale simulation (MSS), is presented in this paper. The basic idea is that the flow is decomposed into several component groups according to spatial and temporal length scales. Each group has its own subdomain, governing system, mesh size, and discretization method. The simulation is then performed groupwise. This approach has been successfully applied in combination with the intergrid dissipation technique for simulation of transitional and turbulent flow in 3-D boundary layers, and it is feasible for 3-D airfoils and other more complex configurations. MSS should prove to ameliorate the scale problems associated with conventional direct numerical simulation.

INTRODUCTION

The main challenge in direct numerical simulation (DNS) is the demand on computer resources. Transitional and turbulent flows contain a wide range of length scales, bounded above by the geometric dimension of the flow field and bounded below by the dissipative action of the molecular viscosity (Canuto et al, 1988). The ratio of the macroscopic (largest) length scale L to the microscopic (smallest) length l (usually called Kolmogorov scale) is $L/l = (Re)^{\frac{3}{4}}$, where Re is the Reynolds number. Thus, for a 3-D problem, the number of grid points, N, must be on the order of $(Re)^{\frac{9}{4}}$ if the Kolmogorov scale is to be resolved. This estimate reveals a fundamental difficulty with DNS for large Reynolds number flows because this resolution requirement is far beyond the capability of current or foreseeable supercomputers. However, this estimate is made based on a single simulation on a single grid and

*Staff Scientist and Associate Professor, Applied Mathematics.

†Assistant Professor Adjunct, Applied Mathematics.

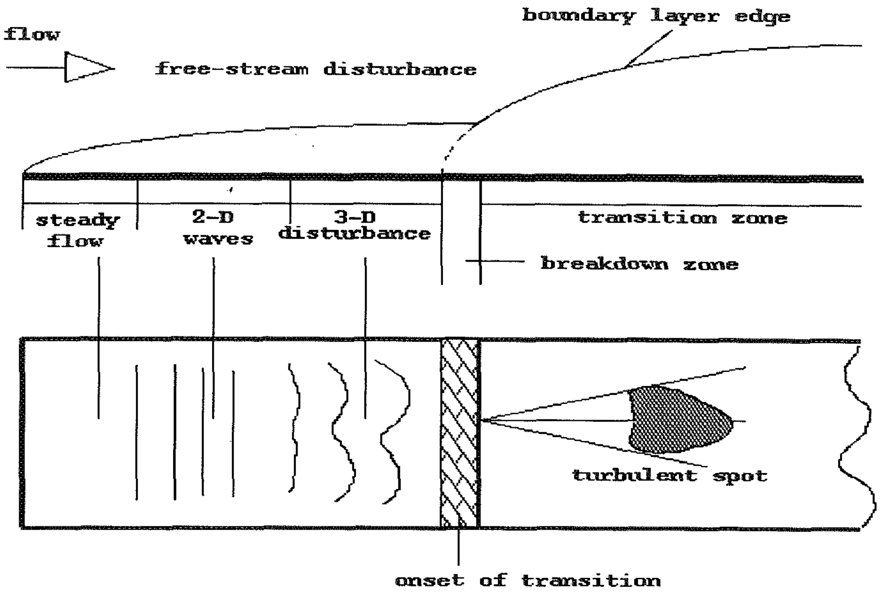


Figure 1. Idealized sketch of transition process on a flat plate.

is, therefore, too pessimistic. Note that the length scales involved in transition and turbulence processes are very different: for an open flow, in general, the main stream and the linear growth of inflow disturbance are dominated by large scales that dominate a large part of the flow field domain; small scales generally occur only in and after breakdown areas. Extremely small scales are only meaningful in a narrow area nearby the solid wall. These observations provide a clue that the total flow may be effectively decomposed into several groups based on their length scales. The large scale flow, dominating most of the flow field, can be simulated by conventional CFD schemes on relatively coarse grids. For small scale flow phenomena, which plays an important role only in a small area of the flow field, high-order discretization and very fine grids have to be used. These small scale simulations may be performed on several grid levels in which each grid has its own subdomain and governing system. This idea eventually leads to a multiple scale simulation (MSS) on several levels of grids. Unlike large eddy simulation (Reynolds, 1990), the MSS approach does not require subgrid models. A basic description of MSS and its performance for CFD problems with simple configuration is the subject of this paper.

ABSTRACT FLOW DECOMPOSITION EXAMPLE

Here we consider the flat plate boundary layer flow as an example to describe the basic idea behind multiple scale simulation. Figure 1 depicts the natural flow transition process in a 3-D boundary layer, showing clearly the variations in flow regime scales.

Using the fact that the flow scale of interest is generally large in the free stream and the area before breakdown (Figure 1), we can consider the use of multiple levels of grid to resolve the flow. Figure 2 depicts the case of three levels used in our boundary layer example; Ω_j represents the domain that level j is used to resolve, with the whole computational domain given by

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3.$$

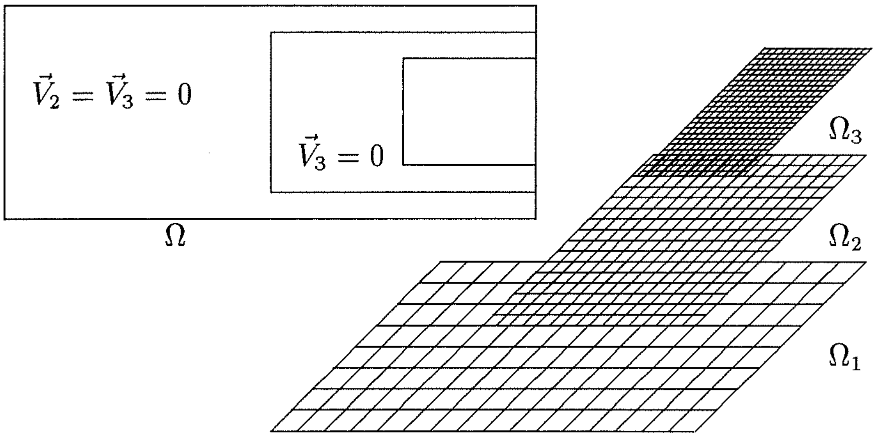


Figure 2. Multiple level grids.

To decompose the total flow according to those levels, suppose the physics is governed by the time-dependent Navier-Stokes equations, which we write as

$$\begin{aligned} &\frac{\partial \vec{V}}{\partial t} + L\vec{V} = \vec{F} && \texttt{in } \Omega, \\ &\vec{V} = \vec{U} + \vec{U}' && \texttt{at inflow.} \end{aligned} \tag{1}$$

Here, we also decompose the inflow vector into two components (usually, $\vec{U}$ is the steady part with large magnitude, and $\vec{U}'$ is the unsteady perturbation part with relatively small magnitude). We then decompose the total flow field into three components according to

$$\vec{V} = \vec{V}_1 + \vec{V}_2 + \vec{V}_3, \tag{2}$$

where $\vec{V}_1$, $\vec{V}_2$, and $\vec{V}_3$ represent increasingly more local and finer scales of the flow so that

$$\begin{aligned} \vec{V}_2 &= 0 && \text{in } \Omega - \Omega_2, \\ \vec{V}_3 &= 0 && \text{in } \Omega - \Omega_3. \end{aligned} \tag{3}$$

To define individual governing systems for each component, first consider the subdomain Ω_1, on which we impose the system

$$\begin{aligned} \frac{\partial \vec{V}_1}{\partial t} + L^{\Omega_1}\vec{V}_1 &= \vec{F}_1 \qquad \texttt{in } \Omega_1, \\ \vec{V}_1 &= \vec{U} \qquad \texttt{at inflow.} \end{aligned} \tag{4}$$

Here, L^{Ω_1} is the spatial difference operator in Ω_1. In general, $\vec{F}_1 \neq \vec{F}$ can be chosen with some freedom to represent large scale physics, so that $\vec{V}_1$ represents the large scale flow without the inflow disturbance. Thus, (4) can generally be solved by low order schemes on a coarse grid. For subdomain Ω_2, we consider the governing system

$$\begin{aligned} \frac{\partial \vec{V}_2}{\partial t} + L^{\Omega_2}(I^{\Omega_2}_{\Omega_1}\vec{V}_1 + \vec{V}_2) &= L^{\Omega_1} I^{\Omega_2}_{\Omega_1}\vec{V}_1 - I^{\Omega_2}_{\Omega_1}\vec{F}_1 + \vec{F} \qquad \texttt{in } \Omega_2, \\ \vec{V}_2 &= \vec{U}' \qquad \texttt{at inflow.} \end{aligned} \tag{5}$$

Here, $I^{\Omega_2}_{\Omega_1}$ represents some interpolation operator to transfer between Ω_1 and Ω_2. Note that $\vec{V}_2$ represents the perturbation in the flow field due to the inflow disturbance $\vec{U}'$ and the presumably finer scale source term $\vec{F} - \vec{F}_1$. $\vec{V}_2$ has a much smaller scale than does $\vec{V}_1$ and should be solved by a high-order scheme (L^{Ω_2}) on a fairly fine middle scale grid. For subdomain Ω_3, which we choose to be a small part of the flow domain, the governing system can be written as

$$\begin{aligned} \frac{\partial \vec{V}_3}{\partial t} + L^{\Omega_3}(I^{\Omega_3}_{\Omega_2} I^{\Omega_2}_{\Omega_1}\vec{V}_1 + I^{\Omega_3}_{\Omega_2}\vec{V}_2 + \vec{V}_3) &= L^{\Omega_2}(I^{\Omega_3}_{\Omega_2} I^{\Omega_2}_{\Omega_1}\vec{V}_1 + I^{\Omega_3}_{\Omega_2}\vec{V}_2) \qquad \texttt{in } \Omega_3, \\ \vec{V}_3 &= 0 \qquad \texttt{on } \partial\Omega_3. \end{aligned} \tag{6}$$

$\vec{V}_3$'s physical scale is considered to be very small so that (6) should be resolved on an extremely fine grid.

Note that (4)–(6) together with the decomposition (2) represent a consistent "lower triangular" formulation that is equivalent to (1) but lends itself to individualized treatment of various physical scales in the discretization. Its triangular form allows for a simplified solution process: first (4) is solved to determine $\vec{V}_1$, then (5) is solved for $\vec{V}_2$, then (6) is solved for $\vec{V}_3$, with the final result then given by $\vec{V} = \vec{V}_1 + \vec{V}_2 + \vec{V}_3$.

APPLICATION TO POISSON EQUATION

The idea of multiple scale simulation as described allows for any desired number of levels, depending on available computer resources and given accuracy requirements. To see the basic process more clearly, we first use a 1-D Poisson equation as an example:

$$\begin{aligned} &\frac{d^2\phi}{dx^2} = -4, \qquad x \in (0,1) \\ &\phi(0) = \phi(1) = 0. \end{aligned} \tag{7}$$

This problem has the analytical solution

$$\phi(x) = 2x(1-x).$$

Using standard central differences for discretization,

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} = f_i,$$

and three levels (Ω_1, Ω_2, Ω_3), we obtain the numerical solution at selected points:

$$\text{in } \Omega_1 \quad \frac{\phi_{1_{i+1}} - 2\phi_{1_i} + \phi_{1_{i-1}}}{.5^2} = -4,$$

$$\text{in } \Omega_2 \quad \frac{\phi_{2_{i+1}} - 2\phi_{2_i} + \phi_{2_{i-1}}}{.25^2} = -4 - (\frac{\bar{\phi}_{1_{i+1}} - 2\bar{\phi}_{1_i} + \bar{\phi}_{1_{i-1}}}{.25^2}),$$

$$\text{in } \Omega_3 \quad \frac{\phi_{3_{i+1}} - 2\phi_{3_i} + \phi_{3_{i-1}}}{.125^2} = -4 - (\frac{\bar{\bar{\phi}}_{1_{i+1}} - 2\bar{\bar{\phi}}_{1_i} + \bar{\bar{\phi}}_{1_{i-1}}}{.125^2} + \frac{\bar{\bar{\phi}}_{2_{i+1}} - 2\bar{\bar{\phi}}_{2_i} + \bar{\bar{\phi}}_{2_{i-1}}}{.125^2}),$$

where $\bar{\phi}_1 = I_{\Omega_1}^{\Omega_2}\phi_1$, $\bar{\bar{\phi}}_1 = I_{\Omega_2}^{\Omega_3}\bar{\phi}_1$, $\bar{\bar{\phi}}_2 = I_{\Omega_2}^{\Omega_3}\phi_2$.

Letting $\phi^{(1)}$, $\phi^{(2)}$, and $\phi^{(3)}$ denote the final solution at grid levels 1, 2, and 3, we obtain the results as shown in Table 1. Obviously, the more the grid levels, the better are the results.

solution	ϕ_1	$\phi^{(1)}$	ϕ_2	$\phi^{(2)}$	ϕ_3	$\phi^{(3)}$	analytical
$\phi(0)$	0	0	0	0	0	0	0
$\phi(0.125)$		0.125		0.1875	0.03125	0.21875	0.21875
$\phi(0.25)$		0.25	0.125	0.375	0.0	0.375	0.375
$\phi(0.375)$		0.375		0.4375	0.03125	0.46875	0.46875
$\phi(0.5)$	0.5	0.5	0	0.5	0.0	0.5	0.5
$\phi(0.625)$		0.375		0.4375	0.03125	0.46875	0.46875
$\phi(0.75)$		0.25	0.125	0.375	0.0	0.375	0.375
$\phi(0.875)$		0.125		0.1875	0.03125	0.21875	0.21875
$\phi(1)$	0	0	0	0	0	0	0

Table 1. Comparison of the numerical solution with three grid levels and the analytical solution for Poisson equation.

This simple example illustrates the basic idea underlying MSS, and it suggests that it might provide a very efficient way to performing DNS for very complex configurations.

FLAT PLATE PROTOTYPE

In this section, we consider spatial flat plate transitional flow as an example to illustrate our approach.

Large Scale Simulation ($\vec{V}_1$)

The governing equation for the base flow is governed by the Navier-Stokes equations. Suppose there is no mass transfer on the flat plate, and gravity is negligible, so that $\vec{F} \equiv 0$. The governing equations can then be written as follows:

$$\begin{aligned} \frac{\partial \vec{V}_1}{\partial t} + (\vec{V}_1 \cdot \nabla)\vec{V}_1 + \nabla P &= \frac{1}{Re}\nabla^2 \vec{V}_1, \\ \nabla \cdot \vec{V}_1 &= 0. \end{aligned} \tag{8}$$

For steady flat plate flow, the Blasius solution can be assumed for the large scale global component:

$$\vec{V}_1 = U_\infty (f'(\eta)\vec{i} + \frac{\eta f'(\eta) - f}{\sqrt{2Re_x}}\vec{j}). \tag{9}$$

where

$$\eta = y\sqrt{\frac{U_\infty}{2\nu x}}, \qquad Re_x = \frac{U_\infty x}{\nu},$$

ν is the kinetic viscosity coefficient, and f can be found in any textbook on boundary layer theory (e.g., Schlichting, 1968).

Middle Scale Simulation ($\vec{V}_2$)

These scales can be determined at inflow for the so-called spatial approach. The governing system is

$$\begin{aligned} \frac{\partial \vec{V}_2}{\partial t} + L^2(\vec{V}_1 + \vec{V}_2) &= L^1 I_{\Omega_1}^{\Omega_2}\vec{V}_1 \qquad \texttt{in } \Omega_2, \\ \vec{V}_2 &= \vec{U}'(t) \qquad \texttt{at inflow}. \end{aligned} \tag{10}$$

Considering $I_{\Omega_1}^{\Omega_2}\vec{V}_1 \equiv (u_1, v_1, w_1, P_1)$ as known, and using (x_1, y_1, z_1, t_1) as the coordinate system on Ω_1, and (x_2, y_2, z_2, t_2) as the coordinate system on Ω_2, then we can write the scalar x-component equation for $\vec{V}_2 \equiv (u_2, v_2, w_2, P_2)$ as

$$\begin{aligned} &\frac{\partial u_2}{\partial t_2} + \frac{\partial (u_1+u_2)(u_1+u_2)}{\partial x_2} + \frac{\partial (u_1+u_2)(v_1+v_2)}{\partial y_2} + \frac{\partial (u_1+u_2)(w_1+w_2)}{\partial z_2} \\ &\qquad + \frac{\partial (P_1+P_2)}{\partial x_2} - \frac{1}{Re}[\frac{\partial^2 (u_1+u_2)}{\partial x_2^2} + \frac{\partial^2 (u_1+u_2)}{\partial y_2^2} + \frac{\partial^2 (u_1+u_2)}{\partial z_2^2}] \\ &= \frac{\partial (u_1 u_1)}{\partial x_1} + \frac{\partial (u_1 v_1)}{\partial y_1} + \frac{\partial (u_1 w_1)}{\partial z_1} - \frac{1}{Re}[\frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_1}{\partial y_1^2} + \frac{\partial^2 u_1}{\partial z_1^2}] + \frac{\partial P_1}{\partial x_1}. \end{aligned} \tag{11}$$

Similarly, the $y-$ and $z-$momentum equations and the continuity equation are:

$$\begin{aligned}
&\frac{\partial v_2}{\partial t_2}+\frac{\partial(u_1+u_2)(v_1+v_2)}{\partial x_2}+\frac{\partial(v_1+v_2)(v_1+v_2)}{\partial y_2}+\frac{\partial(v_1+v_2)(w_1+w_2)}{\partial z_2}\\
&\quad+\frac{\partial(P_1+P_2)}{\partial y_2}-\frac{1}{Re}\Big[\frac{\partial^2(v_1+v_2)}{\partial x_2^2}+\frac{\partial^2(v_1+v_2)}{\partial y_2^2}+\frac{\partial^2(v_1+v_2)}{\partial z_2^2}\Big]\\
&=\frac{\partial(v_1u_1)}{\partial x_1}+\frac{\partial(v_1v_1)}{\partial y_1}+\frac{\partial(v_1w_1)}{\partial z_1}-\frac{1}{Re}\Big[\frac{\partial^2 v_1}{\partial x_1^2}+\frac{\partial^2 v_1}{\partial y_1^2}+\frac{\partial^2 v_1}{\partial z_1^2}\Big]+\frac{\partial P_1}{\partial y_1}, \qquad (12)
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial w_2}{\partial t_2}+\frac{\partial(u_1+u_2)(w_1+w_2)}{\partial x_2}+\frac{\partial(v_1+v_2)(w_1+w_2)}{\partial y_2}+\frac{\partial(w_1+w_2)(w_1+w_2)}{\partial z_2}\\
&\quad+\frac{\partial(P_1+P_2)}{\partial z_2}-\frac{1}{Re}\Big[\frac{\partial^2(w_1+w_2)}{\partial x_2^2}+\frac{\partial^2(w_1+w_2)}{\partial y_2^2}+\frac{\partial^2(w_1+w_2)}{\partial z_2^2}\Big]\\
&=\frac{\partial(u_1w_1)}{\partial x_1}+\frac{\partial(v_1w_1)}{\partial y_1}+\frac{\partial(w_1w_1)}{\partial z_1}-\frac{1}{Re}\Big[\frac{\partial^2 w_1}{\partial x_1^2}+\frac{\partial^2 w_1}{\partial y_1^2}+\frac{\partial^2 w_1}{\partial z_1^2}\Big]+\frac{\partial P_1}{\partial z_1}, \qquad (13)
\end{aligned}$$

$$\frac{\partial u_2}{\partial x_2}+\frac{\partial v_2}{\partial y_2}+\frac{\partial w_2}{\partial z_2}=-\Big[\frac{\partial u_1}{\partial x_1}+\frac{\partial v_1}{\partial y_1}+\frac{\partial w_1}{\partial z_1}\Big]. \qquad (14)$$

Since linear growth and secondary instability are present, $\vec{V_2}$ contains a wide range of differing length scales, some of them rather small. We thus need to use a high-order difference scheme on relatively fine grids. For our purposes, a fourth-order central difference scheme on a staggered grid of resolution $h = O(0.1\lambda)$ is used, where λ is the so-called T-S wavelength.

For a generic partial differential equation,

$$\frac{\partial \phi}{\partial t}+L\phi=S, \qquad (15)$$

we use a second (or higher)-order backward Euler difference in the time direction:

$$\frac{\partial \phi}{\partial t}=\frac{3\phi_{ijk}^{n+1}-4\phi_{ijk}^{n}+\phi_{ijk}^{n-1}}{2\Delta t}+O(\Delta t^2). \qquad (16)$$

Letting $L\phi = (L_h\phi)_{ijk}^{n+1}$, where L_h is the spatial discretization of L described below, yields a fully implicit time-stepping scheme. This has much better stability than the explicit scheme and is much more efficient for representing the nonlinear N-S system. However, it requires solving a large algebraic system at each time step for which we have developed a multigrid algorithm based on so-called line-distributive relaxation (Liu & Liu, 1993). Only one multigrid V-cycle is usually needed to solve this large system, making each implicit time step comparable in CPU cost to a few steps of the corresponding explicit scheme.

To minimize numerical viscosity and phase error, fourth-order central differences (under staggered grid frame) in space is applied:

$$\frac{\partial \phi}{\partial x}|_i \approx \frac{-\phi_{i+2}+8\phi_{i+1}-8\phi_{i-1}+\phi_{i-2}}{12\Delta x},$$

$$
\begin{aligned}
\frac{\partial^2\phi}{\partial x^2}\Big|_i &\approx \frac{-\phi_{i+2}+16\phi_{i+1}-30\phi_i+16\phi_{i-1}-\phi_{i-2}}{12\Delta x^2}, \\
\frac{\partial\phi}{\partial x}\Big|_{i+\frac{1}{2}} &\approx \frac{-\phi_{i+2}+27\phi_{i+1}-27\phi_i+\phi_{i-1}}{24\Delta x}.
\end{aligned}
\tag{17}
$$

Figure 3 depicts the stencil of the discretized x−momentum equation for the interior grid points. (For simplicity, we drop the subscript "2" in Figures 3 and 4.)

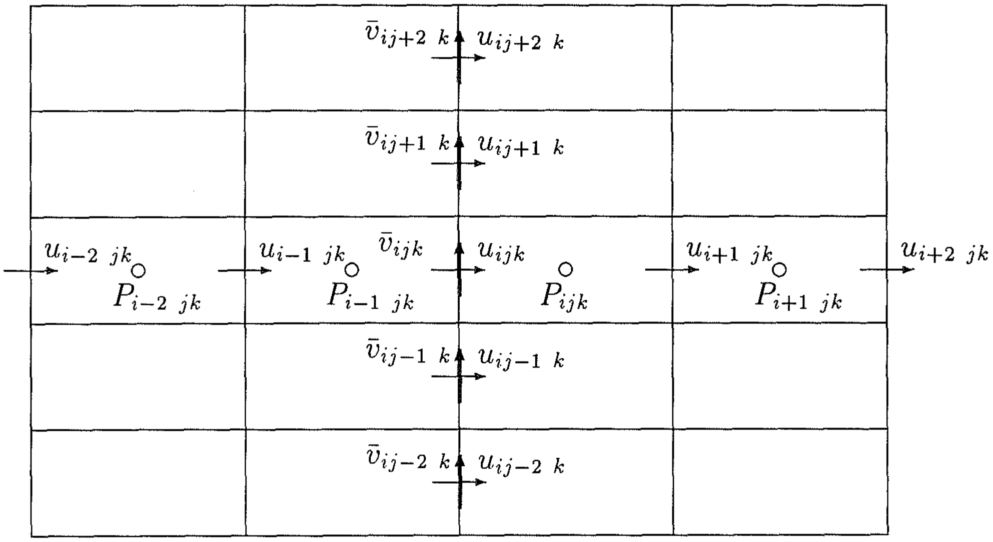


Figure 3. Neighbor points for the x−momentum equation in the (x, y) plane.

Since a staggered grid is used when we discretize the x−momentum equation, we need to evaluate v at the points associated with u where we have no definition for v. This we do by high-order interpolation (Figure 4):

$$
\begin{aligned}
\bar{v}_{ijk} = {} & [9(v_{ijk}+v_{ij+1\ k}+v_{i-1\ jk}+v_{i-1\ j+1\ k}) \\
& -(v_{i-2\ j-1\ k}+v_{i-2\ j+2\ k}+v_{i+1\ j-1\ k}+v_{i+1\ j+2\ k})]/32.
\end{aligned}
\tag{18}
$$

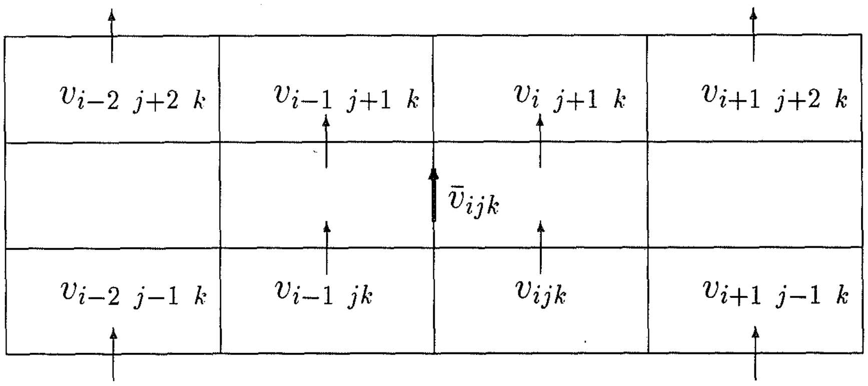


Figure 4. Fourth-order approximation for V_{ijk} at U_{ijk} point.

Small Scale Simulation ($\vec{V}_3$) and Intergrid Dissipation

The subdomain Ω_3 that supports $\vec{V}_3$ includes the transition zones and near wall areas that exhibit very small length scales corresponding to vortex breakdown and transition processes. Very fine grids must therefore be used to resolve these scales. Fortunately, the task that this represents is substantially reduced by the small size of Ω_3 (and, perhaps, the fact that the boundary conditions for $\vec{V}_3$ are homogeneous Dirichlet).

Let I_i^j denote the interpolation operator for the variables transferring from Ω_i to Ω_j, and define

$$\begin{aligned}\bar{u}_0 &= I_2^3(I_1^2 u_1 + u_2), \qquad \bar{v}_0 = I_2^3(I_1^2 v_1 + v_2),\\ \bar{w}_0 &= I_2^3(I_1^2 w_1 + w_2), \qquad \bar{P}_0 = I_2^3(I_1^2 P_1 + P_2).\end{aligned}$$

Furthermore, since the time scale in Ω_3 is also much smaller, we need to obtain the variables at local time $\tilde{t}$. For example,

$$u_0(\tilde{t}) = \frac{\bar{u}_0(t_2^1)\cdot(t_2^2-\tilde{t}) + \bar{u}_0(t_2^2)\cdot(\tilde{t}-t_2^1)}{t_2^2-t_2^1},$$

where t_2^1 and t_2^2 are two time levels in Ω_2. Then the resulting governing system for $\vec{V}_3$ can be written as

$$\begin{aligned}&\frac{\partial u_3}{\partial t_3} + \frac{\partial}{\partial x_3}[(2u_0+u_3)u_3] + \frac{\partial}{\partial y_3}[(v_0+v_3)u_3] + \frac{\partial}{\partial y_3}(v_3u_0)\\ &+\frac{\partial}{\partial z_3}[(w_0+w_3)u_3] + \frac{\partial}{\partial z_3}(w_3u_0) - \frac{1}{Re}(\frac{\partial^2 u_3}{\partial x_3^2} + \frac{\partial^2 u_3}{\partial y_3^2} + \frac{\partial^2 u_3}{\partial z_3^2}) + \frac{\partial P_3}{\partial x_3} \quad =\\ &-[\frac{\partial u_0}{\partial t_2} + \frac{\partial u_0u_0}{\partial x_2} + \frac{\partial u_0v_0}{\partial y_2} + \frac{\partial u_0w_0}{\partial z_2} - \frac{1}{Re}(\frac{\partial^2 u_0}{\partial x_2^2} + \frac{\partial^2 u_0}{\partial y_2^2} + \frac{\partial^2 u_0}{\partial z_2^2}) + \frac{\partial P_0}{\partial x_2}],\end{aligned} \tag{19}$$

$$\begin{aligned}&\frac{\partial v_3}{\partial t_3} + \frac{\partial}{\partial x_3}[(u_0+u_3)v_3] + \frac{\partial}{\partial x_3}(u_3v_0) + \frac{\partial}{\partial y_3}[(2v_0+v_3)v_3]\\ &+\frac{\partial}{\partial z_3}[(w_0+w_3)v_3] + \frac{\partial}{\partial z_3}(w_3v_0) - \frac{1}{Re}(\frac{\partial^2 v_3}{\partial x_3^2} + \frac{\partial^2 v_3}{\partial y_3^2} + \frac{\partial^2 v_3}{\partial z_3^2}) + \frac{\partial P_3}{\partial y_3} \quad =\\ &-[\frac{\partial v_0}{\partial t_2} + \frac{\partial u_0v_0}{\partial x_2} + \frac{\partial v_0v_0}{\partial y_2} + \frac{\partial w_0v_0}{\partial z_2} - \frac{1}{Re}(\frac{\partial^2 v_0}{\partial x_2^2} + \frac{\partial^2 v_0}{\partial y_2^2} + \frac{\partial^2 v_0}{\partial z_2^2}) + \frac{\partial P_0}{\partial y_2}],\end{aligned} \tag{20}$$

$$\begin{aligned}&\frac{\partial w_3}{\partial t_3} + \frac{\partial}{\partial x_3}[(u_0+u_3)w_3] + \frac{\partial}{\partial x_3}(u_3w_0) + \frac{\partial}{\partial y_3}[(v_0+v_3)w_3]\\ &+\frac{\partial}{\partial y_3}(v_3w_0) + \frac{\partial}{\partial z_3}[(2w_0+w_3)w_3] - \frac{1}{Re}(\frac{\partial^2 w_3}{\partial x_3^2} + \frac{\partial^2 w_3}{\partial y_3^2} + \frac{\partial^2 w_3}{\partial z_3^2}) + \frac{\partial P_3}{\partial z_3} \quad =\\ &\frac{\partial w_0}{\partial t_2} + \frac{\partial u_0w_0}{\partial x_2} + \frac{\partial v_0w_0}{\partial y_2} + \frac{\partial w_0w_0}{\partial z_2} - \frac{1}{Re}(\frac{\partial^2 w_0}{\partial x^2} + \frac{\partial^2 w_0}{\partial y^2} + \frac{\partial^2 w_0}{\partial z^2}) + \frac{\partial P_0}{\partial z},\end{aligned} \tag{21}$$

$$\frac{\partial u_3}{\partial x_3} + \frac{\partial v_3}{\partial y_3} + \frac{\partial w_3}{\partial z_3} = -[\frac{\partial u_0}{\partial x_2} + \frac{\partial v_0}{\partial y_2} + \frac{\partial w_0}{\partial z_2}]. \tag{22}$$

The basic approach we use in Ω_3 here is the same as we use in Ω_2. The grids are now much finer, though not yet fine enough to resolve the Kolmogorov scale. Since the central difference scheme is nondissipative, trouble occurs in the breakdown stage where the shear layer develops and the large vortices decompose into small scale ones. The numerical simulation will thus have a huge energy burst, the disturbance velocity will be amplified by several orders of magnitude somewhere inside the flow field, and the computation then goes unstable. These nonphysical phenomena occur because our scheme is nondissipative, and the grid size is not small enough to represent the dissipative small vortices.

The recently developed technique of intergrid dissipation (Liu & Liu, 1994b) can be used to provide the dissipation contributed by small vortices without distortion of the physical solution. We describe this process as follows. At each time step, we make the replacement

$$\vec{V}_3^h \longleftarrow (1-\alpha)\vec{V}_3^h + \alpha I_{2h}^h I_h^{2h} \vec{V}_3^h.$$

Here, the scripts h and $2h$ indicate the respective fine and coarse grid approximations, I_h^{2h} and I_{2h}^h refer to respective restriction and interpolation, and α is a dynamic weight factor. In Ω_3, we choose

$$\alpha = \frac{\Delta x_3 \Delta y_3 \Delta z_3}{\Delta t_3}(\vec{V}_3 \cdot \vec{V}_3), \tag{23}$$

where Δx_3, Δy_3, and Δz_3 are the local spacing in the $x-$, $y-$, and $z-$directions, and Δt_3 is the local time step.

Numerical Test

For the actual computation, a stretched grid that becomes much denser near the solid wall is used. Consider the transformation

$$\begin{aligned} x &= \xi, \\ y &= y(\eta) = \frac{y_{max}\sigma\eta}{\eta_{max}\sigma + y_{max}(\eta_{max} - \eta)}, \\ z &= \zeta, \\ J &= \left|\frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)}\right| = \eta_y, \end{aligned}$$

and

$$\begin{aligned} U &\equiv y_\eta u, \\ V &\equiv v, \\ W &\equiv y_\eta w, \end{aligned}$$

where y_{max} is the height of the computational domain in the physical coordinate y, η_{max} is the height of the computational domain in the computational coordinate η, and σ is a constant that can be used to adjust the concentration of grid points. We can then write the contravariant based governing equations on Ω_3 as follows:

$$\begin{aligned}
&\frac{\partial U_3}{\partial t}+\frac{1}{y_\eta}\frac{\partial}{\partial\xi}[(2U_0+U_3)U_3]+\frac{\partial}{\partial\eta}[\frac{(V_0+V_3)U_3}{y_\eta}]+\frac{\partial}{\partial\eta}[\frac{V_3U_0}{y_\eta}]\\
&+\frac{1}{y_\eta}\frac{\partial}{\partial\zeta}[(W_0+W_3)U_3]+\frac{1}{y_\eta}\frac{\partial}{\partial\zeta}[W_3U_0]+y_\eta\frac{\partial P_3}{\partial\xi}\\
&-\frac{1}{Re_0^*}[\frac{\partial^2U_3}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2}{\partial\eta^2}(\frac{U_3}{y_\eta})+y_\eta\eta_{yy}\frac{\partial}{\partial\eta}(\frac{U_3}{y_\eta})+\frac{\partial^2U_3}{\partial\zeta^2}]\\
&=-\{\frac{\partial U_0}{\partial t}+\frac{1}{y_\eta}\frac{\partial U_0U_0}{\partial\xi}+\frac{\partial}{\partial\eta}[\frac{U_0V_0}{y_\eta}]+\frac{1}{y_\eta}\frac{\partial}{\partial\zeta}[U_0W_0]+y_\eta\frac{\partial P_0}{\partial\xi}\\
&-\frac{1}{Re_0^*}[\frac{\partial^2U_0}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2}{\partial\eta^2}(\frac{U_0}{y_\eta})+y_\eta\eta_{yy}\frac{\partial}{\partial\eta}(\frac{U_0}{y_\eta})+\frac{\partial^2U_0}{\partial\zeta^2}]\},
\end{aligned}\tag{24}$$

$$\begin{aligned}
&y_\eta\frac{\partial V_3}{\partial t}+\frac{\partial}{\partial\xi}[(U_0+U_3)V_3]+\frac{\partial}{\partial\xi}(U_3V_0)+\frac{\partial}{\partial\eta}[(2V_0+V_3)V_3]\\
&+\frac{\partial}{\partial\zeta}[(W_0+W_3)V_3]+\frac{\partial}{\partial\zeta}(W_3V_0)+\frac{\partial P_3}{\partial\eta}\\
&-\frac{1}{Re_0^*}[y_\eta\frac{\partial^2V_3}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2V_3}{\partial\eta^2}+y_\eta\eta_{yy}\frac{\partial V_3}{\partial\eta}+y_\eta\frac{\partial^2V_3}{\partial\zeta^2}]\\
&=-\{y_\eta\frac{\partial V_0}{\partial t}+\frac{\partial U_0V_0}{\partial\xi}+\frac{\partial V_0V_0}{\partial\eta}+\frac{\partial W_0V_0}{\partial\zeta}+\frac{\partial P_0}{\partial\eta}\\
&-\frac{1}{Re_0^*}[y_\eta\frac{\partial^2V_0}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2V_0}{\partial\eta^2}+y_\eta\eta_{yy}\frac{\partial V_0}{\partial\eta}+y_\eta\frac{\partial^2V_0}{\partial\zeta^2}]\},
\end{aligned}\tag{25}$$

$$\begin{aligned}
&\frac{\partial W_3}{\partial t}+\frac{1}{y_\eta}\frac{\partial}{\partial\xi}[(U_0+U_3)W_3]+\frac{1}{y_\eta}\frac{\partial}{\partial\xi}[W_0U_3]+\frac{\partial}{\partial\eta}[\frac{(V_0+V_3)W_3}{y_\eta}]\\
&+\frac{\partial}{\partial\eta}[\frac{V_3W_0}{y_\eta}]+\frac{1}{y_\eta}\frac{\partial}{\partial\zeta}[(2W_0+W_3)W_3]+y_\eta\frac{\partial P_3}{\partial\zeta}\\
&-\frac{1}{Re_0^*}[\frac{\partial^2W_3}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2}{\partial\eta^2}(\frac{W_3}{y_\eta})+y_\eta\eta_{yy}\frac{\partial}{\partial\eta}(\frac{W_3}{y_\eta})+\frac{\partial^2W_3}{\partial\zeta^2}]\\
&=-\{\frac{\partial W_0}{\partial t}+\frac{1}{y_\eta}\frac{\partial U_0W_0}{\partial\xi}+\frac{\partial}{\partial\eta}[\frac{V_0W_0}{y_\eta}]+\frac{1}{y_\eta}\frac{\partial}{\partial\zeta}[W_0W_0]+y_\eta\frac{\partial P_0}{\partial\zeta}\\
&-\frac{1}{Re_0^*}[\frac{\partial^2W_0}{\partial\xi^2}+\frac{1}{y_\eta}\frac{\partial^2}{\partial\eta^2}(\frac{W_0}{y_\eta})+y_\eta\eta_{yy}\frac{\partial}{\partial\eta}(\frac{W_0}{y_\eta})+\frac{\partial^2W_0}{\partial\zeta^2}]\},
\end{aligned}\tag{26}$$

$$\frac{\partial U_3}{\partial\xi}+\frac{\partial V_3}{\partial\eta}+\frac{\partial W_3}{\partial\zeta}=-\{\frac{\partial U_0}{\partial\xi}+\frac{\partial V_0}{\partial\eta}+\frac{\partial W_0}{\partial\zeta}\}.\tag{27}$$

For the details about discretization of the above system, see Liu & Liu (1994a).

To investigate the efficiency of our MSS approach, we choose to investigate the secondary instability case with $Re_0^* = 900$. As above, we use only three levels to describe the flow. A $130 \times 18 \times 10$ grid is employed for both Ω_1 and Ω_2, which includes a 7 T-S wavelength physical domain and a 1 T-S wavelength buffer (Liu & Liu, 1993); a $42 \times 18 \times 18$ patch is used for Ω_3. The patch covers the downstream half of the flat plate except for the buffer domain. The stretch parameter is $\sigma = 3.75$.

As mentioned above, the Blasius similarity solution is employed as the base flow ($\vec{V}_1$), which is widely used as the base flow for flat plate transition. A Benney-Lin type disturbance (Benney & Lin, 1960),

$$U'^{(k)}(0,y,z,t) = \mathrm{Real}\{\epsilon_{2d}\phi_{2d}^{(k)}(y)e^{-i\omega_{2d}t} + \epsilon_{3d+}\phi_{3d+}^{(k)}e^{-i\omega_{3d}t+i\beta z} + \epsilon_{3d-}\phi_{3d-}^{(k)}e^{-i\omega_{3d}t-i\beta z}\},$$

is imposed on the inflow to generate $\vec{V}_2$. Here, $\phi_{2d}(y)$ and $\phi_{3d\pm}(y)$ correspond, respectively, to 2-D and 3-D eigensolutions of the Orr-Sommerfeld equation, and the superscript (k) denotes different velocity components. Other parameters used in this work are as follows:

$$\begin{aligned}
Re_0^* &= 900, \quad Fr = 86 \quad (\omega_{2d} = \omega_{3d} = 0.0774),\\
\beta &= 0.1, \quad y_{max} = \eta_{max} = 50,\\
\alpha_{2d} &= 0.2229 - 0.00451i,\\
\alpha_{3d} &= 0.2169 - 0.00419i,\\
\epsilon_{2d} &= 0.03, \quad \epsilon_{3d\pm} = 0.01,\\
x_0^* &= 303.9, \quad x_{end}^* = 529.4,\\
\Delta t &= T_{T-S}/240.
\end{aligned}$$

Figure 5 depicts the contour plots of the spanwise perturbation vorticity ($\vec{V}_2$) in plane $y_0^* = 0.1123$ at $t = 3T,\ 4T, \cdots, 7T$, where T is the so-called T-S period. It is quite clear that within this level, the flow scale is still pretty large, and only large scale lambda waves can be resolved.

Figure 6 presents contour plots of spanwise vorticity produced by $\vec{V}_3$ in the same plane and at the same time as Figure 5. Though this level is still not fine enough to catch all of the scales in the flow field, some finer scales are resolved. We find that, in the patch (Ω_3), more vortices are generated on this level and they are amplified when they travel downstream. This is at least qualitatively correct.

The final results produced by $\vec{V}_2 + \vec{V}_3$ are described in Figures 7 and 8, showing clearly that more physical details can be found than in Figure 5.

CONCLUDING REMARKS

We have demonstrated the potential of multiscale simulation for solving fluid flow problems to greater resolution and with better efficiency than conventional fixed-scale methods provide. However, several important improvements need to be achieved:

- The 'one-way' refinement approach should be improved by 'two-way' grid processing so that the finer scale resolution more effectively influences the global coarser scales. This would be more in the spirit of a true multilevel algorithm.
- The treatment of the artificial local-grid boundaries should be improved by other than homogeneous Dirichlet conditions to achieve better conservation.
- The local source terms should somehow be improved to provide more accurate fine-scale features.
- The intergrid dissipation scheme plays an important role in allowing the simulation to retain relatively coarse resolution, but the particular choice of the weights here is somewhat ad hoc. We may need to find a more physically based rationale for determining these weights.

ACKNOWLEDGMENTS

The authors thank Professor S. McCormick for his helpful discussions.

REFERENCES

[1] Benney, D.J. & Lin, C.C., 'On the Secondary Motion Induced by Oscillations in a Shear Flow', *Phys. Fluids* **3**, 656, 1960.

[2] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.

[3] Liu, C. and Liu, Z., High order finite difference and multigrid methods for spatially-evolving instability. *J. Comput. Phys.*, **106**, 92-100, 1993.

[4] Liu, Z., and Liu, C., Fourth Order Finite Difference and Multigrid Methods for Modeling Instabilities in Flat Plate Boundary Layers–2-D and 3-D Approaches. *Computers Fluids* **23** No.7, pp.955-982, 1994a.

[5] Liu, C., and Liu, Z., Multigrid mapping and box relaxation for simulation of the whole process of flow transition in 3-D boundary layers, *UCD/CCM Report* **12**, 1994b.

[6] Reynolds, W.C., The potential and limitations of direct and large eddy simulations. In *Whither Turbulence? Turbulence at the Crossroads.* ed. J. L. Lumley, 313-342, Springer–Verlag, 1990.

[7] Schlichting, H., *Boundary Layer Theory*, McGraw-Hill Inc., New York, 1968.

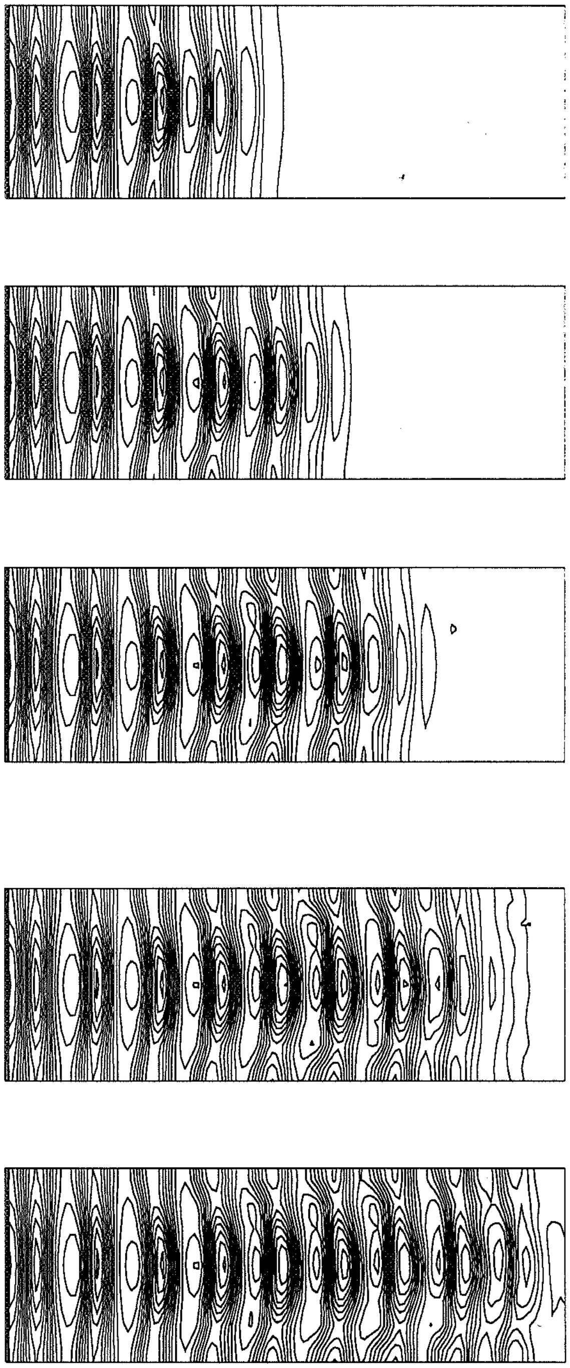


Figure 6. Contour plots of the spanwise vorticity produced by $\vec{V}_2$ in plane $y_0^* = 0.1123$ at $t = 3T,\ 4T, \cdots, 7T$ (from top to bottom).

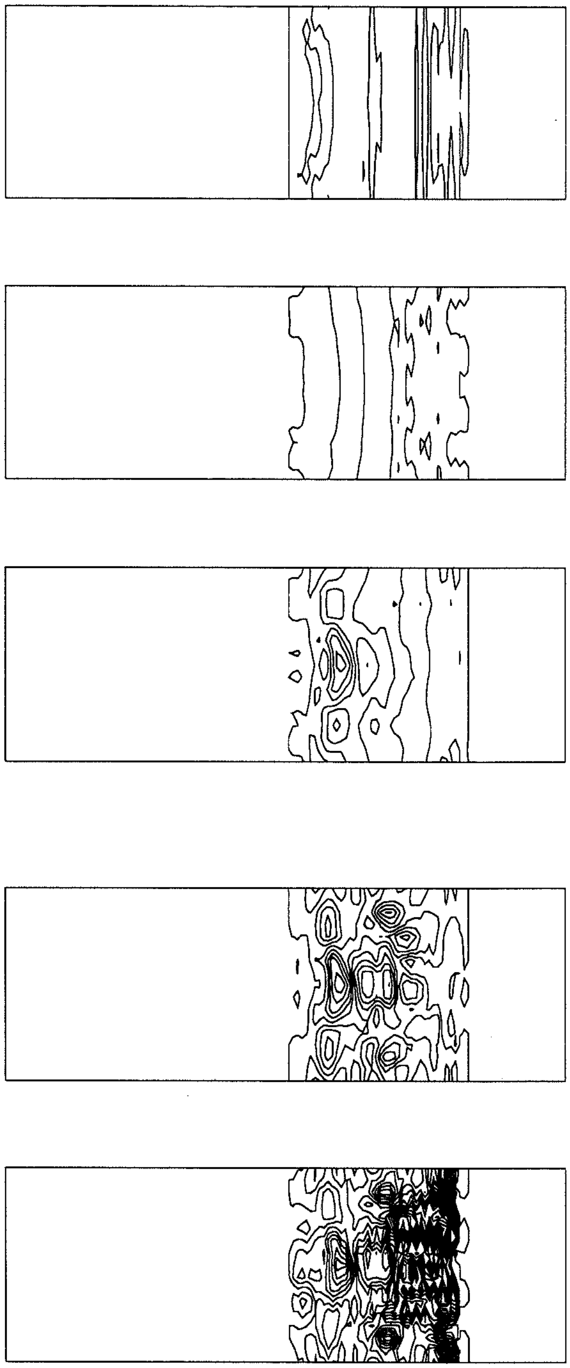


Figure 7. Contour plots of the spanwise vorticity produced by $\vec{V}_3$ in plane $y_0^* = 0.1123$ at $t = 3T,\ 4T, \cdots, 7T$ (from top to bottom).

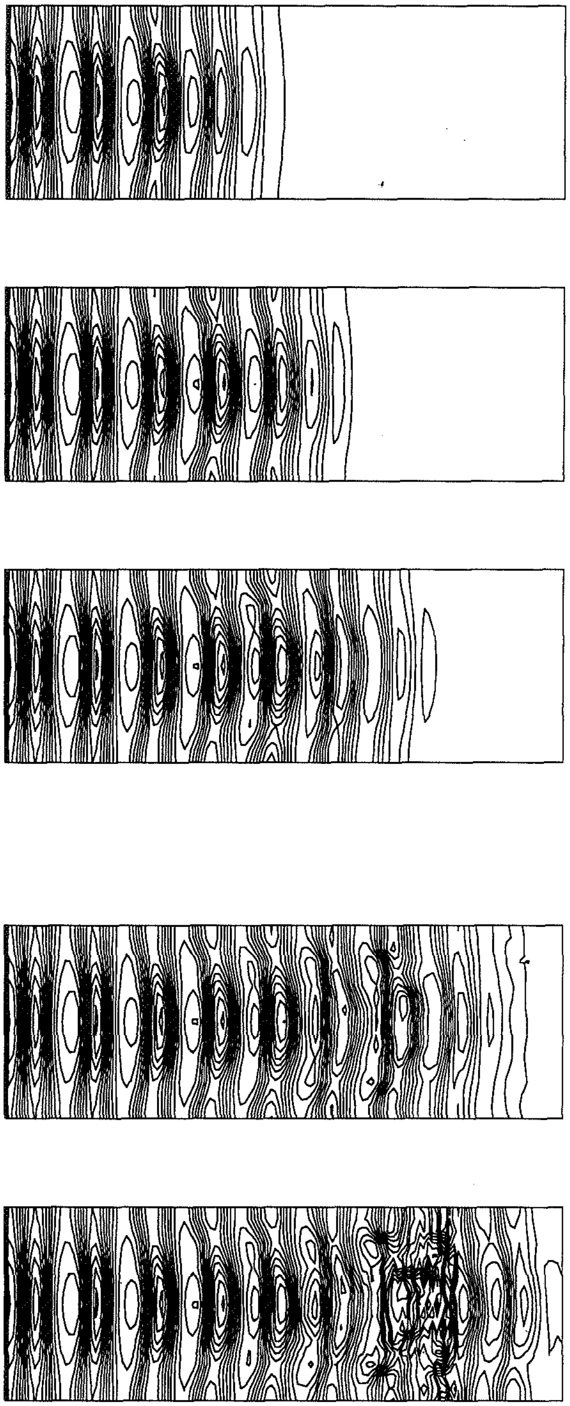


Figure 8. Contour plots of the spanwise vorticity produced by $\vec{V}_2 + \vec{V}_3$ in plane $y_0^* = 0.1123$ at $t = 3T,\ 4T, \cdots, 7T$ (from top to bottom).

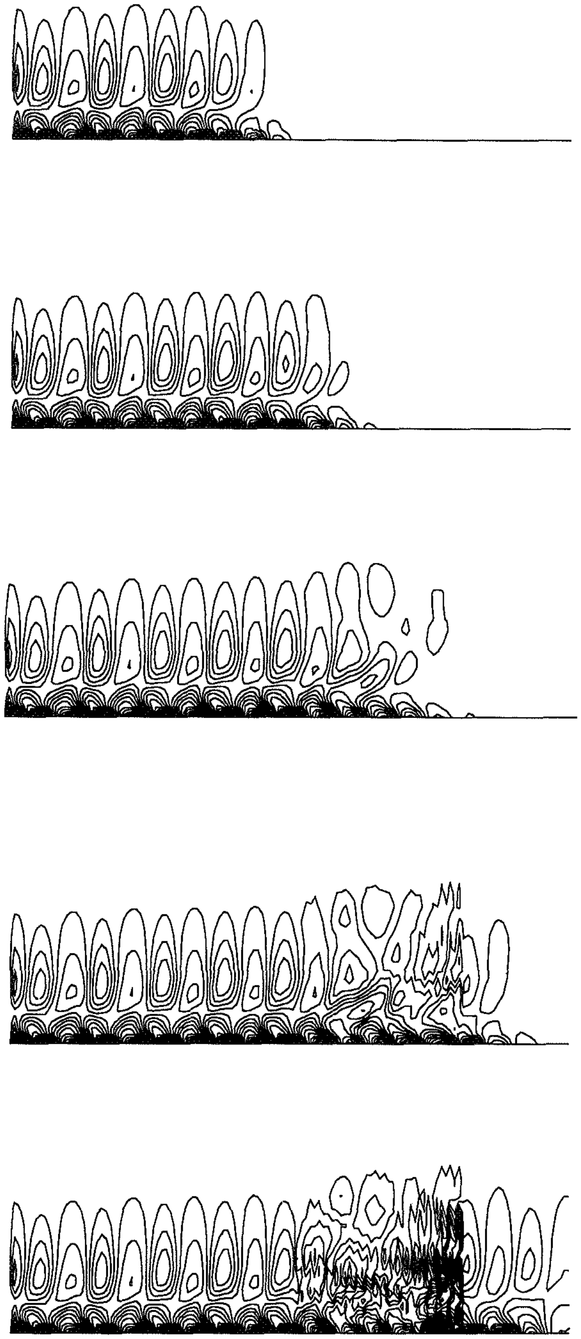


Figure 9. Contour plots of the spanwise vorticity produced by $\vec{V}_2 + \vec{V}_3$ in plane $z_0^* = 0$ at $t = 3T,\ 4T, \cdots, 7T$ (from top to bottom).

Page intentionally left blank

A NOTE ON SUBSTRUCTURING PRECONDITIONING FOR NONCONFORMING FINITE ELEMENT APPROXIMATIONS OF SECOND ORDER ELLIPTIC PROBLEMS

Serguei Maliassov*

SUMMARY

In this paper an algebraic substructuring preconditioner is considered for nonconforming finite element approximations of second order elliptic problems in 3D domains with a piecewise constant diffusion coefficient. Using a substructuring idea and a block Gauss elimination, part of the unknowns is eliminated and the Schur complement obtained is preconditioned by a spectrally equivalent very sparse matrix. In the case of quasiuniform tetrahedral mesh an appropriate algebraic multigrid solver can be used to solve the problem with this matrix. Explicit estimates of condition numbers and implementation algorithms are established for the constructed preconditioner. It is shown that the condition number of the preconditioned matrix does not depend on either the mesh step size or the jump of the coefficient. Finally, numerical experiments are presented to illustrate the theory being developed.

1. INTRODUCTION

Let Ω be a convex bounded domain in $\mathbb{R}^3$ with boundary $\partial\Omega$. Consider an elliptic problem

$$\begin{array}{ll} -\nabla \cdot (k \cdot \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_0, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \Gamma_1, \end{array} \tag{1}$$

where $k(\mathbf{x})$ is a uniformly positive bounded function, $f(\mathbf{x}) \in L^2(\Omega)$, $\overline{\Gamma_0 \cup \Gamma_1} = \partial\Omega$, $\Gamma_0 \cap \Gamma_1 = \emptyset$, and $\Gamma_0 \equiv \overline{\Gamma_0} \neq \emptyset$.

Note that an approach considered in this paper is valid also for the case of the Neumann problem, i.e. $\Gamma_0 = \emptyset$, and it is not described here only for the sake of simplicity.

Let the bilinear form $a(\cdot,\cdot)$ be defined by

$$a(u,v) = (k \cdot \nabla u, \nabla v), \qquad u,v \in V_0(\Omega) = \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_0\},$$

*Institute for Scientific Computation and Department of Mathematics, Texas A&M University, 326 Teague Research Center, College Station, TX 77843–3404. e-mail: *malyasov@isc.tamu.edu*

where $(\cdot, \cdot)$ denotes the $L^2(\Omega)$ inner product. Then the usual weak form of (1) for the solution $u \in V_0(\Omega)$ is

$$a(u, v) = (f, v), \qquad \forall v \in V_0(\Omega). \tag{2}$$

Let $\mathcal{T}_T$ be a regular partitioning of Ω into simplexes T with a mesh size h and let $V_h(\Omega)$ be the P_1–nonconforming finite element space of functions $v \in L^2(\Omega)$ [1] such that $v|_T$ are linear for all $T \in \mathcal{T}_T$ and v are continuous at the barycenters of $T \in \mathcal{T}_T$ and vanish at the barycenters of the boundary faces on Γ_0. Note that the space $V_h(\Omega)$ is not a subspace of $H^1(\Omega)$.

Define the bilinear form on $V_h(\Omega)$ by

$$a_h(u, v) = \sum_{T \in \mathcal{T}_T} (k \cdot \nabla u, \nabla v)_T, \qquad \forall\ u, v \in V_h(\Omega), \tag{3}$$

where $(\cdot, \cdot)_T$ is the $L^2(T)$ inner product, $T \in \mathcal{T}_T$. Then the P_1–nonconforming finite element discretization of (1) is to find $u_h \in V_h$ such that

$$a_h(u_h, v) = (f, v), \qquad \forall v \in V_h(\Omega). \tag{4}$$

Once a basis $\{\varphi_i(\mathrm{x})\}_{i=1}^N$ for $V_h(\Omega)$ is chosen, (4) leads to a system of linear algebraic equations. Write $u(\mathrm{x}) = \sum_{i=1}^N \ u_i \varphi_i(\mathrm{x})$. Then (4) becomes

$$\sum_{i=1}^N u_i a_h(\varphi_i, \varphi_j) = (f, \varphi_j), \qquad j = 1, \ldots, N,$$

or in matrix representation

$$A\mathrm{u} = \mathbf{f}, \tag{5}$$

where $A_{ji} = a_h(\varphi_i, \varphi_j)$, $f_j = (f, \varphi_j)$, $i, j = 1, \ldots, N$.

The first efficient solvers for nonconforming finite element approximations were proposed and investigated in [1] and [2]. Further developments can be found in [3], [4], and [5].

In this paper we will describe and analyze a method of constructing the preconditioner for (5) using an idea of algebraic substructuring (see [6] and [7]), which consists of the following main steps.

First, we represent the matrix A from (5) in a 2×2 block form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{6}$$

where $A_{ii} : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$, $i = 1, 2$, $N_1 + N_2 = N$, in such a way that the block A_{22} is easily invertible. With the introduction of the Schur complement $\hat{A}_{11} = A_{11} - A_{12} A_{22}^{-1} A_{21}$, the matrix A can be rewritten in the form

$$A = \begin{bmatrix} \hat{A}_{11} + A_{12} A_{22}^{-1} A_{21} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{7}$$

Then, we reconstruct the directed graph of the Schur complement $\hat{A}_{11}$ in such a way that the resulting matrix S has the same kernel, is still positive definite (or positive semidefinite if the matrix A is singular), and is spectrally equivalent to the matrix $\hat{A}_{11}$, i.e.,

$$c_0(S\mathbf{u},\mathbf{u}) \leq (\hat{A}_{11}\mathbf{u},\mathbf{u}) \leq c_1(S\mathbf{u},\mathbf{u}), \qquad \forall \mathbf{u} \in \mathbb{R}^{N_1},$$

with constants c_0 and c_1 independent of the mesh step size h and the jump of the coefficient $k(\mathbf{x})$.

To precondition the matrix S we make the same steps. That is, the matrix S is represented in a 2×2 block form

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \tag{8}$$

where $S_{ii} : \mathbb{R}^{N_{1i}} \to \mathbb{R}^{N_{1i}}$, $i = 1, 2$, $N_{11} + N_{12} = N_1$, in such a way that block S_{22} is easily invertible, so that Schur complement $\hat{S}_{11} = S_{11} - S_{12}S_{22}^{-1}S_{21}$ is easily computable.

Finally, following the ideas in [8], [9], and [10], we construct matrix $\tilde{S}_{11}$ spectrally equivalent to $\hat{S}_{11}$ with constants $0 < d_0 \leq d_1$ independent of the mesh size parameter h and the jump of the coefficient $k(\mathbf{x})$:

$$d_0(\tilde{S}_{11}\mathbf{v},\mathbf{v}) \leq (\hat{S}_{11}\mathbf{v},\mathbf{v}) \leq d_1(\tilde{S}_{11}\mathbf{v},\mathbf{v}), \qquad \forall \mathbf{v} \in \mathbb{R}^{N_{11}}.$$

Then the matrix

$$B = \begin{bmatrix} \begin{bmatrix} \tilde{S}_{11} + S_{12}S_{22}^{-1}S_{21} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} + A_{12}A_{22}^{-1}A_{21} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{9}$$

is spectrally equivalent to the matrix A, i.e.,

$$r_0(B\mathbf{u},\mathbf{u}) \leq (A\mathbf{u},\mathbf{u}) \leq r_1(B\mathbf{u},\mathbf{u}), \qquad \forall \mathbf{u} \in \mathbb{R}^{N},$$

where $r_0 = \min\{1; c_0\} \cdot \min\{1; d_0\}$, $r_1 = \max\{1; c_1\} \cdot \max\{1; d_1\}$. In the case of quasiuniform mesh and piecewise constant coefficient $k(\mathbf{x})$, an algebraic multigrid method (AMG) [11], [4], [9], [10] can be used to construct such a matrix $\tilde{S}_{11}$.

In other words the reconstruction of the directed graph of the matrix is equivalent to constructing the equivalent norm on finite dimensional space. An implementation of this approach depends on the structure of the graph of matrix A and, consequently, on the type of nonconforming finite element space V_h. A detailed description of constructing algebraic substructuring preconditioners for one concrete case of the P_1-nonconforming space V_h was given in [12], [13], and [14]. In all these papers authors defined the partitioning $\mathcal{T}_h$ of the whole domain by subdividing it into topological parallelepipeds and splitting each parallelepiped in turn into **six** tetrahedra. The present paper extends these results to the case of splitting each topological parallelepiped into **five** tetrahedra.

The explicit bounds of the spectrum of the preconditioned matrix are obtained with the help of the superelement analysis [12], [10], [7], [15].

The outline of the reminder of the paper is as follows. In Section 2 we consider a formulation of the model problem with piecewise coefficient $k(\mathbf{x})$ when Ω is a unit cube. Then, in Section 3 we develop an algebraic substructuring preconditioner for the resulting linear system and give an implementation algorithm. In Section 4 we outline the algebraic multigrid method we use to precondition the Schur complement obtained in Section 3. Finally, the results of the numerical experiments and some conclusions are given in Section 5 to illustrate the theory being presented.

2. PROBLEM FORMULATION

To explain our approach we consider the model case when Ω is a unit cube in $\mathbb{R}^3$, the boundary conditions are uniform, and $k(\mathbf{x})$ is a piecewise constant function. Note that an extension of the method for the case of Ω being a union of parallelepipeds is straightforward.

Let $\mathcal{C}_h = \{C^{(i,j,k)}\}$ be a partition of Ω into uniform cubes with the length of the edge $h = 1/n$, where (x_i, y_j, z_k) is the right back upper corner of the cube $C^{(i,j,k)}$. Next, divide each cube $C^{(i,j,k)}$ into 5 tetrahedra as shown in Figure 1 and denote this partitioning of Ω into tetrahedra by $\mathcal{T}_h$. Note that we have two types of the partitioning of the cubes $C^{(i,j,k)}$ into tetrahedra and the cube with one type of partitioning has all adjacent cubes of another type. Below we assume that function $k(\mathbf{x})$ is a constant on each cube $C \in \mathcal{C}_h$.

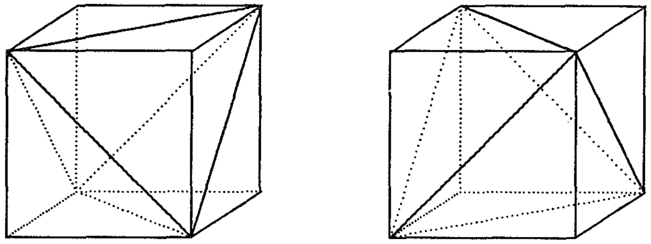


FIGURE 1. *Partition of cubes $C^{(i,j,k)}$ into tetrahedra.*

We introduce the set of barycenters of all faces of the tetrahedral partition of Ω and the set Q_h of those barycenters not on Γ_0. The Crouzeix-Raviart P_1–nonconforming finite element space V_h is defined by

$$V_h = \{v \in L^2(\Omega) : \ v|_T \in P_1(T),\ \forall T \in \mathcal{T}_h;\ v \text{ is continuous at the barycenters from } Q_h \text{ and vanishes at the barycenters of faces on } \Gamma_0\}. \tag{10}$$

Let its dimension be N. Note that $N \approx 10n^3$.

Now we define the bilinear form on V_h by

$$a_h(u,v) = \sum_{T \in \mathcal{T}_h} \int_T k(\mathbf{x}) \nabla u \cdot \nabla v \, d\mathbf{x}, \qquad \forall\, u, v \in V_h. \tag{11}$$

Thus the nonconforming discretization of the problem (1) is given by seeking $u_h \in V_h$ such that

$$a_h(u_h, v) = (f, v), \qquad \forall\, v \in V_h. \tag{12}$$

For any function $v_h \in V_h$ we denote by $\mathbf{v} \in \mathbb{R}^N$ the corresponding vector of its degrees of freedom.

Let $(\mathbf{u}, \mathbf{v})_N$ be a standard bilinear form defined on $\mathbb{R}^N$ by $(\mathbf{u}, \mathbf{v})_N = \sum_{\mathbf{x} \in Q_h} u(\mathbf{x}) v(\mathbf{x})$, $\forall u, v \in V_h$. Then the discretization operator $A : \mathbb{R}^N \to \mathbb{R}^N$, which is symmetric and positive definite, is defined by

$$(A\mathbf{u}, \mathbf{v})_N = a_h(u, v), \qquad u, v \in V_h. \tag{13}$$

Similarly, we introduce the vector $\mathbf{f}$ by $(f, v) = (\mathbf{f}, \mathbf{v})_N$, $\forall\, v \in V_h$. Now, problem (12) can be rewritten in a matrix form

$$A\mathbf{u} = \mathbf{f}. \tag{14}$$

For each cube $C = C^{(i,j,k)} \in \mathcal{C}_h$, denote by V_h^C the subspace of the restriction of the functions in V_h into C. For each $v \in V_h^C$, we indicate by $\mathbf{v}_c$ the corresponding vector. The dimension of V_h^C is denoted by N_c. Obviously, for a cube without faces on Γ_0 we have $N_c = 16$.

The local stiffness matrix A^C on a cube $C \in \mathcal{C}_h$ is given by

$$(A^C \mathbf{u}_c, \mathbf{v}_c)_{N_c} = \sum_{T \subset C} (k(\mathbf{x}) \nabla u_h, \nabla v_h)_T, \qquad \forall u_h, v_h \in V_h^C. \tag{15}$$

Note that matrices A^C are positive definite when $C \cap \Gamma_0 \neq 0$ and semidefinite otherwise. The global stiffness matrix is determined by assembling the local stiffness matrices:

$$(A\mathbf{u}, \mathbf{v})_N = \sum_{C \in \mathcal{C}_h} (A^C \mathbf{u}_c, \mathbf{v}_c)_{N_c}, \qquad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^N. \tag{16}$$

3. ALGEBRAIC SUBSTRUCTURING PRECONDITIONER OVER A CUBE

In this section we construct the algebraic substructuring preconditioner outlined in the Introduction. Toward the end of the section, we divide all unknowns in the system into two groups:

1. The first group consists of

(a) one unknown per cube corresponding to the 1st face of those tetrahedra that are internal for each cube $C \in \mathcal{C}_h$ (see Figure 2, face 1).

(b) all unknowns corresponding to faces of the cubes in the partition $\mathcal{C}_h$, without the faces on Γ_0 (Figure 2, faces $2, 3, \ldots, 13$).

2. The second group consists of the unknowns corresponding to the faces of the tetrahedra that are internal for each cube and that are not in the 1st group (these are unknowns on faces 14, 15, and 16 in Figure 2).

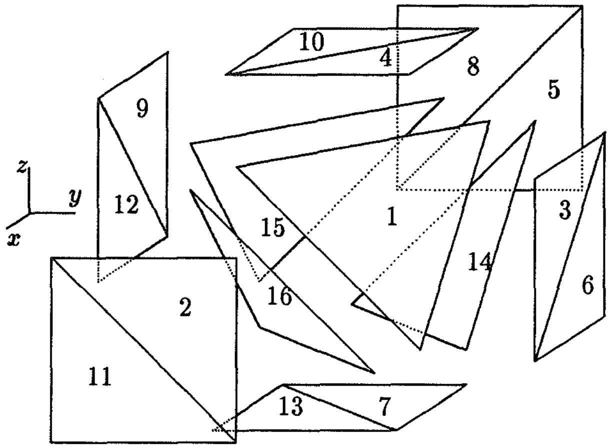


FIGURE 2. *Local enumeration of faces in a cube.*

The splitting of the space $\mathbb{R}^N$ induces the presentation of the vectors $\mathrm{v}^T = (\mathrm{v}_1^T, \mathrm{v}_2^T)$, where $\mathrm{v}_1 \in \mathbb{R}^{N_1}$ and $\mathrm{v}_2 \in \mathbb{R}^{N_2}$, and v_2 corresponds to the unknowns of the second group. Obviously, $N_2 = 3n^3$ and $N_1 = N - 3n^3$. Then the matrix A can be presented in the following block form:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{17}$$

where $A_{ii} : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$, $i = 1, 2$.

Note that the matrix A_{22} is block diagonal and can be inverted locally (cube by cube). Thus, Schur complement $\hat{A}_{11} = A_{11} - A_{12}A_{22}^{-1}A_{21}$ is easily computable.

The local stiffness matrices on each cube also have the block form:

$$A^C = \frac{3h}{2} k_c \begin{bmatrix} A_{11,c} & A_{12,c} \\ A_{21,c} & A_{22,c} \end{bmatrix}, \tag{18}$$

where $A_{22,c}$ are 3×3 matrices.

An important fact which is established by direct computations is that the matrix $\hat{A}_{11}$ can be obtained by assembling over all cubes local matrices $\hat{A}_{11,c} = A_{11,c} -$

$A_{12,c}A_{22,c}^{-1}A_{21,c}$:

$$(\hat{A}_{11}\mathbf{u}_1, \mathbf{v}_1) = \sum_{C\in\mathcal{C}_h} \frac{3h}{2} k_c(\hat{A}_{11,c}\mathbf{u}_{1,c}, \mathbf{v}_{1,c}), \qquad \forall \mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{N_1}.$$

Here $\mathbf{u}_{1,c}$ is a restriction of $\mathbf{u}_1$ into the nodes of the first group on the cube $C \in \mathcal{C}_h$, and for the cube $C \in \mathcal{C}_h$ without faces on Γ_0 we have $\dim \mathbf{u}_{1,c} = 13$.

Let us consider a cube C that has no face on the boundary Γ_0 and enumerate the faces s_j, $j = 1, \ldots, 16$, as shown on Figure 2. Then the local matrices $A_{ij,c}$, $i, j = 1, 2$, of this cube have the following form:

$$A_{11,c} = \begin{bmatrix} 9/2 & -\mathbf{r}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{r} & I & & & \\ \mathbf{0} & & I & & \\ \mathbf{0} & & & I & \\ \mathbf{0} & & & & I \end{bmatrix}, \qquad A_{22,c} = \frac{1}{2}\begin{bmatrix} 9 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & 9 \end{bmatrix}, \qquad (19)$$

$$A_{12,c}^T = A_{21,c} = \begin{bmatrix} -1/2 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix},$$

where $\mathbf{r} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$, and I is 3×3 identical matrix.

The local Schur complement matrix $\hat{A}_{11,c}$ for this cube has the form

$$\hat{A}_{11,c} = \frac{1}{7} \cdot \begin{bmatrix} 30 & -7\mathbf{r}^T & -\mathbf{r}^T & -\mathbf{r}^T & -\mathbf{r}^T \\ -7\mathbf{r} & 7I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{r} & \mathbf{0} & T & -R & -R \\ -\mathbf{r} & \mathbf{0} & -R & T & -R \\ -\mathbf{r} & \mathbf{0} & -R & -R & T \end{bmatrix}, \qquad (20)$$

where

$$T = \frac{1}{5} \cdot \begin{bmatrix} 27 & -8 & -8 \\ -8 & 27 & -8 \\ -8 & -8 & 27 \end{bmatrix}, \qquad R = \frac{1}{5} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Along with the matrix $\hat{A}_{11,c}$ we introduce on each cube $C \in \mathcal{C}_h$ the 13×13 matrices S_c by

$$S_c = \begin{bmatrix} 12 & -\mathbf{r}^T & -\mathbf{r}^T & -\mathbf{r}^T & -\mathbf{r}^T \\ -\mathbf{r} & I & & & \\ -\mathbf{r} & & I & & \\ -\mathbf{r} & & & I & \\ -\mathbf{r} & & & & I \end{bmatrix}, \qquad (21)$$

and define $N_1 \times N_1$ matrix S by assembling over all cubes local matrices S_c:

$$(S\mathbf{u}_1, \mathbf{v}_1) = \sum_{C\in\mathcal{C}_h} \frac{3h}{2} k_c(S_c\mathbf{u}_{1,c}, \mathbf{v}_{1,c}), \qquad \forall \mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{N_1}.$$

It is easy to see that the matrices $\hat{A}_{11,c}$ and S_c have the same kernel, i.e., $\ker\hat{A}_{11,c} = \ker S_c$.

We now consider an eigenvalue problem for $\mu \neq 0$:

$$\hat{A}_{11,c}\mathbf{u} = \mu S_c\mathbf{u}, \qquad \mathbf{u} \neq 0, \qquad \mathbf{u} \in \mathbb{R}^{13}. \tag{22}$$

Direct calculations show that the eigenvalues of this problem belong to the interval $[\mu_{min}, \mu_{max}]$, where $\mu_{min} = 1/7$ and $\mu_{max} = 1$.

Defining a new $N_c \times N_c$ matrix on each cube

$$B^C = \frac{3h}{2}k_c \begin{bmatrix} S_c + A_{12,c}A_{22,c}^{-1}A_{21,c} & A_{12,c} \\ A_{21,c} & A_{22,c} \end{bmatrix}, \tag{23}$$

we define the symmetric positive-definite $N \times N$ matrix B by

$$(B\mathbf{u}, \mathbf{v}) = \sum_{C \in \mathcal{C}_h} (B^C\mathbf{u}_c, \mathbf{v}_c), \tag{24}$$

where $\mathbf{v}, \mathbf{u} \in \mathbb{R}^N$, and $\mathbf{u}_c$ and $\mathbf{v}_c$ are their respective restrictions on the cube C.

To estimate the condition number of the matrix $B^{-1}A$ we use so called superelement analysis (see [16], [9], [17], [7]). Namely, it is easy to show the following inequalities:

$$\max_{(B\mathbf{u},\mathbf{u})\neq 0} \frac{(A\mathbf{u}, \mathbf{u})}{(B\mathbf{u}, \mathbf{u})} = \max_{(B\mathbf{u},\mathbf{u})\neq 0} \frac{\sum\limits_{C\in\mathcal{C}_h} (A^C\mathbf{u}_c, \mathbf{u}_c)}{\sum\limits_{C\in\mathcal{C}_h} (B^C\mathbf{u}_c, \mathbf{u}_c)} \leq \max_{\substack{C\in\mathcal{C}_h \\ (B^C\mathbf{u}_c,\mathbf{u}_c)\neq 0}} \frac{(A^C\mathbf{u}_c, \mathbf{u}_c)}{(B^C\mathbf{u}_c, \mathbf{u}_c)} \tag{25}$$

and

$$\min_{(B\mathbf{u},\mathbf{u})\neq 0} \frac{(A\mathbf{u}, \mathbf{u})}{(B\mathbf{u}, \mathbf{u})} = \min_{(B\mathbf{u},\mathbf{u})\neq 0} \frac{\sum\limits_{C\in\mathcal{C}_h} (A^C\mathbf{u}_c, \mathbf{u}_c)}{\sum\limits_{C\in\mathcal{C}_h} (B^C\mathbf{u}_c, \mathbf{u}_c)} \geq \min_{\substack{C\in\mathcal{C}_h \\ (B^C\mathbf{u}_c,\mathbf{u}_c)\neq 0}} \frac{(A^C\mathbf{u}_c, \mathbf{u}_c)}{(B^C\mathbf{u}_c, \mathbf{u}_c)}. \tag{26}$$

From the inequalities (25) and (26) we see that to estimate the condition number of $B^{-1}A$, it is sufficient to consider the local eigenvalue problems for $\mu_c \neq 0$ on each cube:

$$A^C\mathbf{u}_c = \mu_c B^C\mathbf{u}_c, \qquad \mathbf{u}_c \neq 0, \qquad \mathbf{u}_c \in \mathbb{R}^{N_c}.$$

From (22) direct calculations show that the eigenvalues μ_c are within the interval $[1/7, 1]$. Then the inequalities (25) and (26) yield:

PROPOSITION 1. *The eigenvalues of the problem* $A\mathbf{u} = \lambda B\mathbf{u}$, $\mathbf{u} \neq 0$, *belong to the interval* $[1/7, 1]$, *and the condition number is thus estimated by* $\mathrm{cond}(B^{-1}A) \leq 7$.

We stress that the condition number of the matrix $B^{-1}A$ is bounded by a constant independent of the mesh step size h and the jump of the coefficient $k(\mathbf{x})$.

Let us take the matrix B from (24) as a preconditioner for the matrix A. In the terms of the group partitioning introduced above it has the following block form

$$B = \begin{bmatrix} S + A_{12}A_{22}^{-1}A_{21} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{27}$$

As we noted earlier, the matrix A_{22} is block-diagonal and can be inverted locally on cubes. So we concentrate on the linear system

$$S\mathbf{w} = \mathbf{G}, \qquad \mathbf{w}, \mathbf{G} \in \mathbb{R}^{N_1}. \tag{28}$$

The matrix S also can be represented in the block form

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \tag{29}$$

where the block S_{22} corresponds to the nodes from the subgroup (b) of the first group, which are on the faces of cubes $C \in \mathcal{C}_h$. From the definition of S, it can be seen that the matrix S_{22} is diagonal. In the above partitioning, we present $\mathbf{w}$ and $\mathbf{G}$ in (29) in the form

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix}, \tag{30}$$

where the dimension of vectors $\mathbf{w}_1$ and $\mathbf{G}_1$ is obviously equal to $M = \dim \mathbf{w}_1 = n^3$ and $\dim \mathbf{w}_2 = N_1 - n^3$. Then, after elimination of the second group of unknowns $\mathbf{w}_2 = S_{22}^{-1}(\mathbf{G}_2 - S_{21}\mathbf{w}_1)$, we get the system of linear equations

$$(S_{11} - S_{12}S_{22}^{-1}S_{21})\mathbf{w}_1 = \mathbf{G}_1 - S_{12}S_{22}^{-1}\mathbf{G}_2 \equiv \tilde{\mathbf{G}}_1, \tag{31}$$

where the vector $\mathbf{w}_1$ and the block S_{11} correspond to the unknowns from the subgroup (a) of the first group, which have only one unknown per cube.

Thus, if we define as above the Schur complement of matrix S by $\hat{S}_{11} = S_{11} - S_{12}S_{22}^{-1}S_{21}$, matrix B can be presented in the form

$$B = \begin{bmatrix} \begin{bmatrix} \hat{S}_{11} + S_{12}S_{22}^{-1}S_{21} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} + A_{12}A_{22}^{-1}A_{21} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{32}$$

where matrix A_{22} is block diagonal and S_{22} is diagonal and can be inverted locally cube-by-cube. Again, we have to stress that the condition number of the matrix $B^{-1}A$ is bounded by the constant independent of the mesh step size h and the jump of the coefficient $k(\mathbf{x})$. The matrix B can be referred to as a three-level preconditioner.

It is easy to see that the Schur complement $\hat{S}_{11}$ is a "7-point-scheme" matrix. In the next section we consider the solution techniques for problem (31) with the matrix $\hat{C}_{11}$.

4. MULTILEVEL PRECONDITIONER OVER A CUBE

While the preconditioner B has good properties, it is still not economical to invert it because the entries of the matrix $\hat{S}_{11}$ depend on the jump of the coefficients. In this section we propose a preconditioner for the matrix $\hat{S}_{11}$ provided that additional assumptions on the behavior of the function $k(\mathbf{x})$ are met and show that for this modification we can use any well-known multilevel procedure.

Assumption (**A1**). Suppose that unit cube Ω can be represented as a union of a certain number m of pairwise disjoint cubes G_i, $i = 1, \ldots, m$, with the size of edge H ($H > 2h$) in such a way that in each cube G_i the function $k(\mathbf{x})$ is a positive constant. In other words, we set $\bar{\Omega} \doteq \bigcup_{i=1}^{m} \bar{G}_i$ and $k(\mathbf{x}) = const_i > 0$, $\mathbf{x} \in G_i$, $i = 1, \ldots, m$.

Now define on Ω an auxiliary parallelepipedal mesh $\tilde{\mathcal{T}}_C$ with vertices in the centers of cubes $C^{(i,j,k)} \in \mathcal{T}_C$ and in the centers of the boundary faces $\partial C^{(i,j,k)} \cap \partial\Omega$. Let us consider a standard partitioning of $\tilde{\mathcal{T}}_C$ into tetrahedra $\tilde{\mathcal{T}}_h$ and enumerate the nodes of this mesh in accordance with the enumeration of the cubes of $\mathcal{T}_C$.

Then define the piecewise constant function $\tilde{k}(\mathbf{x})$ to be constant on each cube $\tilde{C}^{(i,j,k)} \in \tilde{\mathcal{T}}_C$ by

$$\tilde{k}(\mathbf{x}) = \min_{\alpha,\beta,\gamma=0,1} \left\{ k^{(i+\alpha,j+\beta,k+\gamma)} \right\}, \qquad \mathbf{x} \in \tilde{C}^{(i,j,k)}, \tag{33}$$

and consider the boundary value problem

$$-\nabla \cdot \left(\tilde{k}\, \nabla u \right) = g \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \Gamma_0. \tag{34}$$

Denote by U_h a usual (conforming) finite element space of all continuous piecewise linear functions on $\tilde{\mathcal{T}}_h$ that vanish at the nodes of Γ_0. Note that $\dim U_h = M$. And, finally, define the symmetric positive definite matrix $\tilde{C}$ by

$$(\tilde{C}\mathbf{u}, \mathbf{v})_M = \int\limits_{\Omega} \tilde{k} \nabla u \cdot \nabla v \, dx \qquad \forall u, v \in U_h, \tag{35}$$

where $\mathbf{u}$, $\mathbf{v}$ are the vectors of degrees of freedom corresponding to the functions u and v, respectively.

Consider an eigenvalue problem

$$\hat{S}_{11}\mathbf{u} = \mu \tilde{C}\mathbf{u}, \qquad \mathbf{u} \neq 0, \qquad \mathbf{u} \in \mathbb{R}^M. \tag{36}$$

The following statement plays a very important role in all further arguments [15]. It can be established by straightforward computations.

PROPOSITION 2. *The eigenvalues of the problem* (36) *belong to the interval* $[1/2, 1]$.

Now instead of the matrix (32) we define new matrix $\tilde{B}$ by

$$\tilde{B} = \begin{bmatrix} \begin{bmatrix} \tilde{C} + S_{12}S_{22}^{-1}S_{21} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} + A_{12}A_{22}^{-1}A_{21} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{37}$$

Then we can formulate the following theorem.

THEOREM 1. *The matrix $\tilde{B}$ defined in* (37) *with the block $\tilde{C}$ defined in* (35) *is spectrally equivalent to the matrix A and* $\mathrm{cond}(\tilde{B}^{-1}A) \leq 14$.

Thus, we have constructed a spectrally equivalent sparse preconditioner for the Schur complement after the elimination of almost 90% of the original unknowns. We note here that matrices A_{22} and S_{22} are block diagonal and, with $\tilde{B}$ as a preconditioner for the matrix A, we have to develop procedure for solving the linear system of equataions

$$\tilde{C}\mathbf{u} = \mathbf{G}, \qquad \mathbf{u} \in \mathbb{R}^M. \tag{38}$$

We have to stress that the function $\tilde{k}(\mathbf{x})$ is piecewise constant. Thus, any multilevel procedure which works well for such problems (34) can be used.

We apply the preconditioned conjugate gradient method to solve the problem (13) with the matrix $\tilde{B}$ from (37) as a preconditioner for the matrix A and use the multilevel domain decomposition method (MGDD) [9], [10], [15] to solve the problem (38) with matrix $\tilde{C}$; we establish the following results.

STATEMENT 1. *If we use the MGDD method to solve problem* (38) *with the matrix $\tilde{C}$, then the condition number* $\mathrm{cond}(\tilde{B}^{-1}A)$ *does not depend on mesh size h and the jump of the coefficient $k(\mathbf{x})$.*

STATEMENT 2. *The number of operations for solving the system $A\mathbf{u} = \mathbf{f}$ by the preconditioned conjugate gradient method with preconditioner $\tilde{B}$ and with accuracy ε in the sense*

$$||\mathbf{u}^{k_\varepsilon+1} - \mathbf{u}^*||_A \leq \varepsilon||\mathbf{u}^0 - \mathbf{u}^*||_A,$$

is estimated by $C \cdot N \cdot \ln\frac{2}{\varepsilon}$, where $\mathbf{u}^* = A^{-1}\mathbf{f}$, $\mathbf{u}^0 \in \mathbb{R}^N$, and C does not depend on N and jump of the coefficient $k(\mathbf{x})$.

5. RESULTS OF THE NUMERICAL EXPERIMENTS

In this section the preconditioner being considered is tested on the model problem

$$\begin{aligned} -\nabla \cdot (k(\mathbf{x})\nabla u) &= f, \qquad \text{in } \Omega = [0,1]^3, \\ u &= 0, \qquad \text{on } \partial\Omega. \end{aligned}$$

In the numerical experiments presented we use the preconditioner $\tilde{B}$ in the form (37). In this case by the Theorem 1 Cond $\tilde{B}^{-1}A \leq 14$. The problem with matrix $\tilde{C}$ is solved by the multilevel domain decomposition method, as described in [15].

The domain is divided into $M = n^3$ cubes (n in each direction) and each cube is partitioned into 5 tetrahedra. The dimension of the original algebraic system is $N = 10n^3 - 6n^2$. The right hand side is generated randomly, and the accuracy parameter is taken as $\varepsilon = 10^{-6}$. The condition numbers of the preconditioned matrices $B^{-1}A$ are calculated by the relation between the conjugate gradient and Lanczos algorithms. The coefficient $k(\mathbf{x})$ is piecewise constant and is defined to be

$$k(x,y,z) = \begin{cases} k, & (x,y,z) \in [0.5,1] \times [0.5,1] \times [0.5,1] \\ 1, & \text{elsewhere} \end{cases} \tag{39}$$

The results are summarized in Table 1, where n_{iter} and *cond* denote the iteration number and condition number, respectively. All experiments are carried out on a Sun workstation. It takes approximately 25 minutes to solve the problem of the largest dimension $N = 1\ 235\ 000$.

From Table 1 we see that the condition number does not depend on either the step mesh size h or the jump of the coefficient $k(\mathbf{x})$.

TABLE 1. *Solving $\tilde{C}$ by MGDD method*

	$20 \times 20 \times 20$ $N = 77\ 600$		$30 \times 30 \times 30$ $N = 264\ 600$		$40 \times 40 \times 40$ $N = 630\ 400$		$50 \times 50 \times 50$ $N = 1\ 235\ 000$	
k	n_{iter}	cond	n_{iter}	cond	n_{iter}	cond	n_{iter}	cond
1	14	5.32	14	5.30	14	5.29	14	5.28
10	17	6.59	17	6.53	16	6.37	16	6.29
100	17	6.94	17	6.90	16	6.89	16	6.88
1000	17	6.98	16	6.96	16	6.95	16	6.93
10^4	16	6.98	16	6.96	16	5.95	16	6.94
0.1	16	5.97	16	5.96	16	5.96	15	5.94
0.01	16	6.02	16	6.02	16	6.00	15	5.97
0.001	16	6.02	16	6.01	16	6.00	15	5.97
10^{-4}	16	6.02	16	6.01	16	6.00	15	5.97

REFERENCES

[1] Arnold, D. and Brezzi, F., Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates, *RAIRO Model. Math. Anal. Numer.*, 19:7–32, 1985.

[2] Brenner, S., An optimal-order multigrid method for P1 nonconforming finite elements, *Math. Comp.*, 52:1–16, 1989.

[3] Braess, D. and Verfürth, R., Multigrid methods for nonconforming finite element methods, *SIAM J. Numer. Anal.*, 27:979–986, 1990.

[4] Bramble, J., Pasciak, J., and Xu, J., The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms, *Math. Comp.*, 56:1–34, 1991.

[5] Chen, Z. and Kwak, D., The analysis of multigrid algorithms for nonconforming and mixed methods for second order elliptic problems, *IMA Preprint #1277*, 1994.

[6] Kuznetsov, Y., Multilevel substructuring preconditioners, Invited presentation at the 7th Int. Symp. on Domain Decomp. Methods for PDEs, October 1993, Penn State University.

[7] Kuznetsov, Y. and Maliassov, S., Substructuring preconditioner for nonconforming finite element approximations of second order elliptic problems with anisotropy, Technical Report ISC-95-01-MATH, Institute for Scientific Computation, Texas A&M University, 1995.

[8] Bramble, J., Pasciak, J., and Schatz, A., The construction of preconditioners for elliptic problems by substructuring, I, *Math. Comp.*, 47:103–134, 1986.

[9] Kuznetsov, Y., Multigrid domain decomposition methods for elliptic problems, *Proc. of 8th Int. Conf. on Comput. Methods in Applied Science and Engineering*, Paris, 1987. Also *Comput. Meth. Appl. Mech and Eng.*, 75:185–193, 1989.

[10] Kuznetsov, Y., Multigrid domain decomposition methods, *Proc. of* 3*rd International Symposium on Domain Decomposition Methods*, (Chan, T., Glowinski, R., Periaux, J., and Widlund, O., eds.), SIAM, Philadelphia, pp. 290–313, 1989.

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

[12] Ewing, R., Kuznetsov, Y., Lazarov, R., and Maliassov, S., Substructuring preconditioning for finite element approximations of second order elliptic problems. I. Nonconforming linear elements for the Poisson equation in parallelepiped, *IMA Preprint #1280*, 1994.

[13] Ewing, R., Kuznetsov, Y., Lazarov, R., and Maliassov, S., Preconditioning of nonconforming finite element approximations of second order elliptic problems, *The Third Int. Conf. on Advances in Numerical Methods and Applications*, (Dimov, I., Sendov, B., and Vassilevski, P., eds.), World Scientific, Bulgaria, pp. 101–110, 1994.

[14] Chen, Z., Ewing, R., Kuznetsov, Y., Lazarov, R., and Maliassov, S., Multilevel preconditioners for mixed methods for second order elliptic problems, *IMA Preprint #1269*, 1994. (Submitted to *SIAM Numer. Anal.*)

[15] Maliassov, S., Substructuring preconditioning for finite element approximations of second order elliptic problems. II. Mixed method for an elliptic operator with scalar tensor, Technical Report ISC-94-19-MATH, Institute for Scientific Computation, Texas A&M University, 1994.

[16] Axelsson, O., On multigrid methods of the two-level type, *Multigrid methods*, (Hackbush, W. and Trottenberg, U., eds.), No.960 in LNM, Springer, Köln-Porz, pp. 352–367, 1981.

[17] Ewing, R., Lazarov, R., and Vassilevski, P., Local refinement techniques for elliptic problems on cell-centered grids, I: Error Analysis, *Math. Comput.*, 56:437–462, 1991.

Page intentionally left blank

CONVERGENCE OF A SUBSTRUCTURING METHOD WITH LAGRANGE MULTIPLIERS[1]

Jan Mandel and Radek Tezaur
Center for Computational Mathematics
University of Colorado at Denver
Denver, CO

SUMMARY

We analyze the convergence of a substructuring iterative method with Lagrange multipliers, proposed recently by Farhat and Roux. The method decomposes finite element discretization of an elliptic boundary value problem into Neumann problems on the subdomains and a coarse problem for the subdomain nullspace components. For linear conforming elements and preconditioning by the Dirichlet problems on the subdomains, we prove the asymptotic bound on the condition number $C(1+\log(H/h))^\gamma$, $\gamma = 2$ or 3, where h is the characteristic element size and H is the subdomain size.

INTRODUCTION

We analyze the convergence of a substructuring method with Lagrange multipliers, proposed by Farhat and Roux [11] under the name Finite Element Tearing and Interconnecting (FETI) method. The main idea of the FETI method is to decompose the problem domain into nonoverlapping subdomains and to enforce continuity on subdomain interfaces by Lagrange multipliers. Eliminating the subdomain variables yields a dual problem for the Lagrange multipliers, which is solved by preconditioned conjugate gradients. This idea is related to the fictitious domain method where the Lagrange multipliers enforce boundary conditions as in Dinh *et al.* [5].

Elimination of the subdomain variables is implemented by solving Neumann problems on all subdomains in every iteration, which can be done completely in parallel. However, the subdomain problems are singular, so a small auxiliary problem for the nullspace components of the subdomain solutions needs to be solved in every iteration. This is an added complication, but also a blessing. Farhat, Mandel, and Roux [10] have shown numerically and have proved for the FETI method without preconditioning that the auxiliary problem plays the role of a coarse problem, namely, it causes the condition number to be bounded independently of the number of

[1] This research was supported by the National Science Foundation under grants ASC-9217394 and ASC-9121431. This paper has been submitted for journal publication elsewhere.

subdomains. The method was further extended to time-dependent problems, which lack the naturally occurring coarse problem. by Farhat. Chen, and Mandel [9].

In this paper. we show that the condition number of the preconditioned FETI method is bounded independently of the number of subdomains and polylogarithmically in terms of subdomain size. as is the case for other optimal nonoverlapping domain decomposition methods [3. 6. 8. 16. 17]. We refer to [10] for numerical results that confirm the theory and for parallel implementation and performance.

The FETI method is in a sense dual to the Neumann-Neumann method with a coarse problem. developed by Mandel under the name Balancing Domain Decomposition [15] based on an earlier method of de Roeck and LeTallec [4]. A modified method was analyzed by Dryja and Widlund [8].

Analysis of domain decomposition methods typically proceeds by demonstrating spectral equivalence of the quadratic form that defines the problem in a variational setting and the quadratic form that defines the preconditioner. often by way of the P. L. Lions lemma [1. 6. 7. 14]. Since the preconditioner in the FETI method is quite complicated and is not defined in terms of a quadratic form. we proceed differently and find a bound on the norm of the product of the system operator and the preconditioner to bound the maximal eigenvalue. as well as a bound on the inverse to bound the minimal eigenvalue. Related analyses were previously done for methods without crosspoints between the subdomains. or done formally in functional spaces (cf., for example, Glowinski and Wheeler [12]). In this paper. we present a complete analysis in terms of upper and lower bounds on the preconditioned operator for decompositions with crosspoints in 2D and edges and crosspoints in 3D.

FORMULATION OF THE METHOD

In this section, we briefly review formulation of the FETI method according to [10], where one can find more details about the algorithmic side. At the same time, we introduce the spaces and operators that will be used in our analysis.

We consider iterative solution of a system of linear equations $Lx = b$ arising from a finite element discretization of an elliptic boundary value problem on a bounded domain Ω, which is decomposed into nonoverlapping subdomains Ω_i, $i = 1, \ldots, n_s$. The matrix A is assumed to be symmetric and positive definite. Let

$$W_i = V_h(\partial\Omega_i) \tag{1}$$

be the space of local vectors of degrees of freedom associated with the boundary of Ω_i, and let

$$Y = V_h(\bigcup_{i=1}^{n_s} \partial\Omega_i) \tag{2}$$

be the space of global vectors of degrees of freedom associated with all subdomain boundaries. The correspondence of the local and global vectors of degrees of freedom is given by zero-one matrices $N_i : W_i \to Y$.

We find it convenient to identify vectors of degrees of freedom, which are in some spaces $\mathbb{R}^n$, with the associated finite element functions. Operators between the spaces are represented as matrices, and we frequently commit an abuse of notations by using matrices and operators interchangeably. The l^2 inner product is denoted by $\langle\cdot,\cdot\rangle$ on all spaces. The associated norm is $\|u\|^2 = \langle u, u\rangle$. The transpose of a matrix M is denoted by M'.

After elimination of the interior degrees of freedom in all subdomains Ω_i, we obtain the reduced system of linear equations for the vectors $w_i \in W_i$ of degrees of freedom on subdomain boundaries, which we write in subassembly form as

$$\sum_{i=1}^{n_s} N_i S_i w_i = f \tag{3}$$

$$\sum_{i=1}^{n_s} B_i w_i = 0 \tag{4}$$

Here, S_i are the Schur complements of the subdomain stiffness matrices obtained by elimination of the interior degrees of freedom, and B_i are matrices with entries $0, 1, -1$ such that (4) expresses the continuity of the solution between subdomains, that is, the requirement that the values of degrees of freedom common to more than one subdomain coincide.

To describe the method in a concise form, we need to define the following spaces. W is a space of all boundary degrees of freedom on all subdomains:

$$W = \bigotimes_{i=1}^{n_s} W_i \tag{5}$$

X is a space of vectors with entries corresponding to pairs of degrees of freedom on the interfaces where we enforce continuity:

$$X \subset \bigotimes_{\partial\Omega_i \cap \partial\Omega_j \neq \emptyset} V_h(\partial\Omega_i \cap \partial\Omega_j). \tag{6}$$

Denote the block matrix

$$B : W \to X = (B_1, \ldots, B_{n_s}) \tag{7}$$

and the space of Lagrange multipliers

$$U = \text{Range } B. \tag{8}$$

These are the details we need for the purpose of describing the method. A more specific description of B will be given in the next section. Finally, denote the symmetric block diagonal matrix

$$S : W \to W, \quad S = \begin{pmatrix} S_1 & 0 & & & 0 \\ 0 & S_2 & 0 & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & \cdot & 0 \\ 0 & & & 0 & S_{n_s} \end{pmatrix} \tag{9}$$

The problem (3) and (4) can now be written as the minimization of total subdomain energy subject to the continuity condition:

$$\mathcal{E}(w) = \frac{1}{2}\langle Sw, w\rangle + \langle f, w\rangle \to \min, \quad \text{subject to} \quad w \in W, \ Bw = 0. \tag{10}$$

Writing the Lagrangean of this minimization problem

$$\mathcal{L}(w, \lambda) = \frac{1}{2}\langle Sw, w\rangle + \langle f, w\rangle + \langle \lambda, Bw\rangle, \quad w \in W, \ \lambda \in U,$$

we solve the dual problem

$$\max_{\lambda \in U} \inf_{w \in W} \mathcal{L}(w, \lambda) \equiv \max_{\lambda \in U} \mathcal{C}(\lambda). \tag{11}$$

By a direct computation,

$$\mathcal{C}(\lambda) = \begin{cases} -\infty & \text{if } \langle f, w\rangle + \langle \lambda, Bw\rangle \neq 0 \ \text{ for some } w \in \operatorname{Ker} S, \\ -\frac{1}{2}\langle S^+(f - B'\lambda), f - B'\lambda\rangle & \text{otherwise,} \end{cases} \tag{12}$$

where $S^+ : W \to W$ is any pseudoinverse of S, i.e., an operator such that $w = S^+ g$ solves $Sw = g$ if $g \perp \operatorname{Ker} S$. It is easy to see from (12) that the choice of S^+ does not change the value of $\mathcal{C}$. Without loss of generality, assume that S^+ is given by the spectral decomposition

$$S^+ = \sum_{t>0} \frac{1}{t} v_t v_t', \tag{13}$$

where

$$S = \sum_t t v_t v_t', \qquad S v_t = t v_t, \quad v_t' v_t = 1. \tag{14}$$

The dual problem (11) is equivalent to maximizing $\mathcal{C}(\lambda)$ on the admissible set

$$\mathcal{A} = \{\lambda \in U \mid \mathcal{C}(\lambda) > -\infty\}.$$

Define the space of admissible increments

$$\begin{aligned} V &= \{\lambda_1 - \lambda_2 \mid \lambda_1 \in \mathcal{A}, \lambda_2 \in \mathcal{A}\} \\ &= \{\mu \in U \mid \langle \mu, Bw\rangle = 0 \quad \forall w \in \operatorname{Ker} S\}. \end{aligned} \tag{15}$$

At the maximum of $\mathcal{C}(\lambda)$, $\lambda \in \mathcal{A}$, the derivative of $\mathcal{C}$ is zero in all directions in V:

$$D\mathcal{C}(\lambda;\mu) = 0 \quad \forall \mu \in V.$$

By a straightforward computation, this becomes

$$\lambda \in \mathcal{A}, \quad \langle -BS^+B'\lambda + BS^+f, \mu\rangle = 0, \quad \forall \mu \in V. \tag{16}$$

In order to express (16) as a linear equation in the space V, let $P_V : U \to V$ be the projection onto V orthogonal in the l_2 inner product $\langle .,.\rangle$. Then for $\mu \in V$,

$$\langle -BS^+B'\lambda + BS^+f, \mu\rangle = \langle -BS^+B'\lambda + BS^+f, P_V\mu\rangle = \langle P_V(-BS^+B'\lambda + BS^+f), \mu\rangle$$

since P_V is orthogonal, so $P_V = P_V'$. Therefore, the dual problem (11) is equivalent to the linear equation in V for the unknown μ:

$$\mu \in V, \qquad P_V(-BS^+B'(\mu + \lambda_0) + BS^+f) = 0, \tag{17}$$

where λ_0 is an arbitrary starting feasible solution, i.e., $\lambda_0 \in \mathcal{A}$.

The FETI method is the method of preconditioned conjugate gradients in the space V applied to the linear equation (17). The linear part of the operator in (17) is P_VF, where

$$F = BS^+B'. \tag{18}$$

We consider the preconditioner P_VM, where

$$M = A'SA, \qquad A = \frac{1}{2}B'. \tag{19}$$

That is, in each iteration of the preconditioned conjugate gradients algorithm, $z = P_VMr$ is evaluated as an approximate solution of the residual equation $P_VFz = r$. The preconditioner (19) was proposed in [10]. Note that the evaluation of the matrix-vector product Su can be implemented by solving a Dirichlet problem in each subdomain; therefore it is called the Dirichlet preconditioner in [10].

ANALYSIS

A well known bound on the reduction of the error in k iterations of the method of preconditioned conjugate gradients in the norm $|||e||| = \langle P_VFe, e\rangle^{1/2}$ on V is [13]

$$2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k,$$

where κ is the condition number

$$\kappa = \frac{\lambda_{\max}(P_VFP_VM|_V)}{\lambda_{\min}(P_VFP_VM|_V)} \tag{20}$$

and $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of operators on V.

Abstract Framework

The main idea of our convergence analysis is summarized in the following lemma, which we will apply to F and M from (18) and (19).

Lemma 1 *Let U be a finite dimensional linear space with the inner product $\langle \cdot, \cdot \rangle$. Let V be a subspace of U, $\| \cdot \|_V$ be a norm on V induced by an inner product, and the dual norm be defined by $\|v\|_{V'} = \sup_{\tilde{v} \in V} \langle v, \tilde{v} \rangle / \|\tilde{v}\|_V$. Let $P_V : U \to V$ be the $\langle \cdot, \cdot \rangle$ orthogonal projection onto V, and $F, M : U \to V$ linear operators symmetric on V,*

$$\begin{aligned} \langle \tilde{\lambda}, F\lambda \rangle &= \langle \lambda, F\tilde{\lambda} \rangle \quad \forall \lambda, \tilde{\lambda} \in V \\ \langle \tilde{v}, Mv \rangle &= \langle v, M\tilde{v} \rangle \quad \forall v, \tilde{v} \in V, \end{aligned}$$

and such that

$$c_1 \|\lambda\|^2_{V'} \leq \langle \lambda, F\lambda \rangle \leq c_2 \|\lambda\|^2_{V'} \qquad \forall \lambda \in V \tag{21}$$

$$c_3 \|v\|^2_V \leq \langle v, Mv \rangle \leq c_4 \|v\|^2_V \qquad \forall v \in V \tag{22}$$

with constants $c_1, c_2, c_3, c_4 > 0$. Then

$$\kappa = \frac{\lambda_{\max}(P_V M P_V F)}{\lambda_{\min}(P_V M P_V F)} \leq \frac{c_2 c_4}{c_1 c_3}. \tag{23}$$

Proof. Since $\lambda \in V$, we can replace in (21) F by $P_V F$. From (21), the operator norm of the mapping $P_V F : V' \to V$ and its inverse satisfies

$$\|P_V F\|_{V' \to V} \leq c_2, \qquad \|(P_V F)^{-1}\|_{V \to V'} \leq \frac{1}{c_1}. \tag{24}$$

Similarly, (22) implies

$$\|P_V M\|_{V \to V'} \leq c_4, \qquad \|(P_V M)^{-1}\|_{V' \to V} \leq \frac{1}{c_3}. \tag{25}$$

Consequently,

$$\lambda_{\max}(P_V M P_V F) \leq \|P_V M P_V F\|_{V' \to V'} \leq c_2 c_4$$

and

$$\lambda_{\max}((P_V F)^{-1}(P_V M)^{-1}) \leq \|(P_V F)^{-1}(P_V M)^{-1})\|_{V' \to V'} \leq \frac{1}{c_1 c_3},$$

which gives (23). □

The rest of this paper is concerned with estimating the condition number κ from (23). We will specify a suitable norm $\| \cdot \|_V$ and estimate the constants in Lemma 1 for the finite element problem below.

Assumptions

We need more specific assumptions in order to be able to prove a bound on the condition number κ. So, we are solving the boundary value problem

$$\mathcal{A}u = g \quad \text{in} \quad \Omega, \quad u = 0 \quad \text{on} \quad \partial\Omega$$

where

$$\mathcal{A}v = -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left(\alpha(x)\frac{\partial v(x)}{\partial x_j}\right),$$

with $\alpha(x)$ a measurable function such that $0 < \alpha_0 \leq \alpha(x) \leq \alpha_1$ a.e. in Ω.

The domain Ω is assumed to be divided into nonoverlapping subdomains Ω_i, $i = 1, ..., n_s$, which can be generated from a reference domain (square or cube) $\hat{\Omega}$ of unit diameter as $\Omega_i = F_i(\hat{\Omega}_i)$ by mappings F_i, which are assumed to satisfy

$$\|\partial F_i\| \leq CH, \qquad \|\partial F_i^{-1}\| \leq CH^{-1}$$

with the Jacobian ∂F_i and the Euclidean $\mathbb{R}^d$ matrix norm $\|.\|$. That is, the subdomains are shape-regular and have a diameter of $O(H)$.

Assume that $V_h(\Omega)$ is a conforming P1 or Q1 finite element space on a triangulation of Ω, which satisfies the standard regularity and inverse assumptions. Denote by h the characteristic element size. Each subdomain Ω_i is assumed to be a union of some of the elements, and all functions in $V_h(\Omega)$ are zero on $\partial\Omega$.

In particular, the degrees of freedom are values at nodes of the triangulation. We assume that B is defined as follows. For a pair of degrees of freedom $w_r(x_\alpha)$ on $\partial\Omega_r$ and $w_s(x_\alpha)$ on $\partial\Omega_s$, such that the node x_α does not belong to any other subdomain, let

$$(Bw)_{rs}(x_\alpha) = \sigma_{rs}(w_r(x_\alpha) - w_s(x_\alpha)), \tag{26}$$

where $\sigma_{rs} = 1$ or $\sigma_{rs} = -1$.

When node x_β belongs to more than two subdomains $\partial\Omega_i, i = s_1, s_2, \ldots, s_{n_\beta}$, we assume that $(Bw)_{rs}(x_\beta)$ is defined so that B is full rank and so that the coefficients are ± 1 and determined uniquely by the indices $(s_1, s_2, \ldots, s_{n_\beta})$. For example,

$$(Bw)_{k,k+1}(x_\beta) = (-1)^k w_{s_k}(x_\beta) - (-1)^k w_{s_{k+1}}(x_\beta), \quad k = 1, .., n_\beta - 1. \tag{27}$$

For an example of the definition of the values of B from (27) with $(s_1, s_2, s_3) = (1, 3, 2)$ in 2D around a crosspoint, see Fig. 1.

Remark 2 *The essential property here is that there are no redundant constraints in enforcing the continuity of the solution at the nodes where more than two subdomains meet and that the constraints do not change along the edges (in 3D). Only the improved estimate in statement 3 of Lemma 8 will require the specific definition (27).*

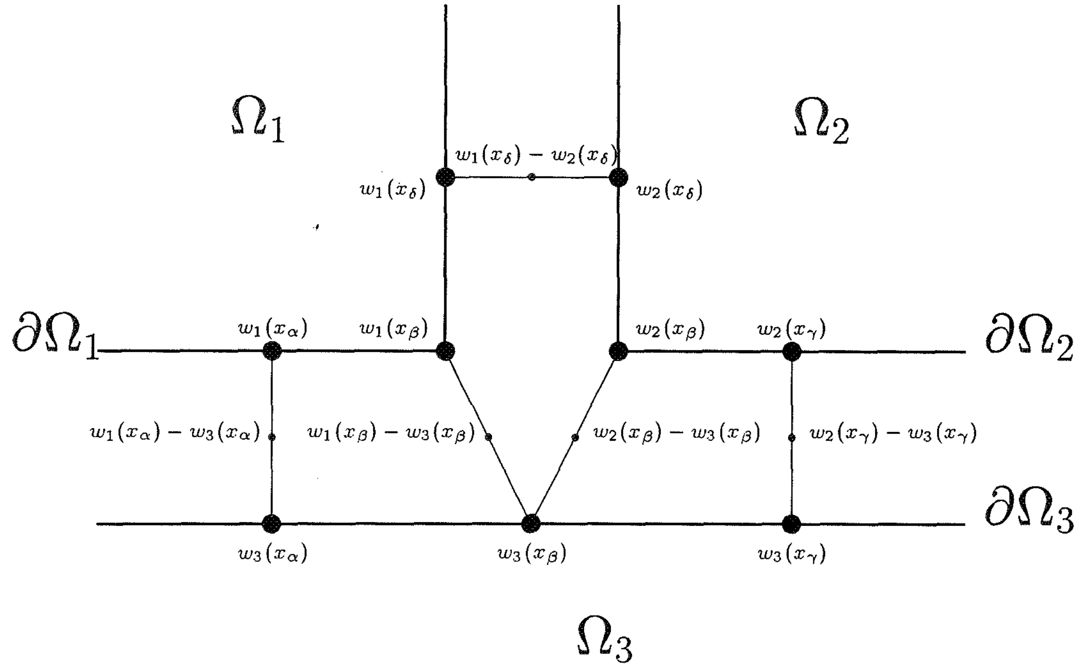


Figure 1: Definition of B

Discrete Norm Bounds

The key to our analysis is a proper choice of norms. We equip the space W with the seminorm and the norm

$$|w|_W^2 = \sum_{i=1}^{n_s} |w_i|^2_{1/2,\partial\Omega_i}, \qquad \|w\|_W^2 = |w|_W^2 + \frac{1}{H}\sum_{i=1}^{n_s} \|w_i\|^2_{0,\partial\Omega_i}, \tag{28}$$

and the space V with the norm $\|\cdot\|_V$ and the dual norm $\|\cdot\|_{V'}$

$$\|v\|_V = \|Av\|_W, \qquad \|v\|_{V'} = \sup_{\tilde{v}\in V} \frac{\langle v, \tilde{v}\rangle}{\|\tilde{v}\|_V}. \quad v \in V. \tag{29}$$

For the definition and properties of the Sobolev seminorms $|\cdot|_{k,O}$, see, e.g., [18]. The space U is identified with some space $\mathbb{R}^n$. We use the l^2 inner product $\langle\cdot,\cdot\rangle$ as duality pairing.

In the following, we use $a \approx b$ to indicate that $ca \leq b \leq Ca$ with some positive generic constants c and C independent of the characteristic mesh size h and the subdomain diameter H. First we need to relate our discrete norm to a Sobolev norm and to establish equivalence of the norm and seminorm on the complement of the kernel of S.

Lemma 3 $|w|_W^2 \approx \langle w, Sw\rangle$, $w \in W$.

Proof. The lemma follows from the standard result [2, 19]

$$|w_i|^2_{H^{1/2}(\partial\Omega_i)} \approx \langle w_i, S_i w_i\rangle$$

by summation over all subdomains Ω_i and using (28). □

Lemma 4 $|w|_W \approx \|w\|_W$, $w \in W$, $w \perp \operatorname{Ker} S$.

Proof. From the equivalence of the H^1 norm and seminorm on the factorspace modulo constants [18] or from the Poincaré inequality, and scaling from a reference domain to subdomain Ω_i,

$$\|w_i\|^2_{0,\partial\Omega_i} \leq CH|w_i|^2_{1/2,\partial\Omega_i}$$

for all w_i if $\partial\Omega_i$ contains a part of $\partial\Omega$, and for all w_i such that $\int_{\partial\Omega_i} w_i = 0$ otherwise. The lemma follows by summation over the subdomains and from (28). □

We also need the equivalence of the norm $\|Av\|_W$ and the seminorm $|Av|_W$.

Lemma 5 $|Av|_W \approx \|Av\|_W$, $v \in V$.

Proof. Let $v \in V$. Since $A = \frac{1}{2}B'$, by definition of V, we have $\langle Av, w\rangle = 0$ $\forall w \in \operatorname{Ker} S$ or $Av \perp \operatorname{Ker} S$, which yields the result using Lemma 4. □

Our norm on V was chosen so that the preconditioner is coercive and bounded, i.e., so that (22) holds with c_1 and c_2 independent of H and h. This is shown in the following lemma.

Lemma 6 $\langle v, Mv\rangle \approx \|v\|^2_V$, $\forall v \in V$,

Proof. For $v \in V$, by definition of the preconditioner M, Lemma 3 and Lemma 5,

$$\langle v, Mv\rangle = \langle v, A'SAv\rangle = \langle Av, SAv\rangle \approx \|v\|_V$$

□

The following lemmas lead to estimates of coercivity and ellipticity of F. We first summarize some well known results and inequalities in a form suitable for our purposes.

Lemma 7 *Let G be a vertex, edge, or face (if $d = 3$) of subdomain Ω_i. A face is understood not to contain adjacent edges, and an edge does not contain its endpoints. For $z \in V_h(\partial\Omega_i)$, define $w \in V_h(\partial\Omega_i)$ by $w(x) = z(x)$ on all nodes $x \in G$; $w(x) = 0$ on all other nodes of $\partial\Omega_i$. Then,*

$$\|w\|^2_{H^{1/2}(\partial\Omega_i)} \leq C(1 + \log\frac{H}{h})^\beta(\|z\|^2_{H^{1/2}(\partial\Omega_i)} + \frac{1}{H}\|z\|^2_{L^2(\partial\Omega_i)}),$$

where

$\beta = 1$ *if* $d = 2$ *and* G *is a vertex, or* $d = 3$ *and* G *is an edge or a vertex*

$\beta = 2$ *if* $d = 2$ *and* G *is an edge, or* $d = 3$ *and* G *is a face.*

Proof. The inequality for $d = 2$ was proved in [16, 17]. The case when $d = 3$ follows from Lemmas 4.1 and 4.2 in [3] if G is an edge or a vertex and Lemma 4.3 in [3] if G is a face (cf. [6]).□

Lemma 8 *It holds that*

$$\inf_{\substack{\tilde{w} \in W \\ B\tilde{w} = Bw}} \|\tilde{w}\|_W^2 \leq C(1 + \log(H/h))^{\alpha} \|ABw\|_W^2, \quad w \in W,$$

where $\alpha = 1$, *and* $\alpha = 0$ *in the following special cases:*

1. $BA = I$, *which means that there are no nodes shared by more than two subdomains.*

2. $d = 2$ *and the matrix* A *has the following property: If* $\bar{w} \in \mathrm{Range}\, A$, x *is a crosspoint (node shared by more than two subdomains), and* $\bar{w}_i(x) = w_i(y)$ *for all* i *such that* $x \in \partial\Omega_i$ *and all nodes* y *that are adjacent to* x *on* $\partial\Omega_i$, *then* $\bar{w}_i(x) = 0$ *for all* i *such that* $x \in \partial\Omega_i$.

3. $d = 2$, B *is defined by (26) and (27), and all nodes in the triangulation belong to either one, two, or an odd number of subdomains.*

Proof. Let us first prove that in the general case we obtain $\alpha \leq 1$. Let $w \in W$ and $u = Bw$ throughout this proof. From the fact that $BA(BA)^{-1}u = u$, and by the triangle inequality,

$$\inf_{\substack{\tilde{w} \in W \\ B\tilde{w} = u}} \|\tilde{w}\|_W \leq \|A(BA)^{-1}u\|_W \leq \|Au\|_W + \|A(I - (BA)^{-1})u\|_W. \tag{30}$$

Denote $z \doteq A(I - (BA)^{-1})u$. From the definition of B in (26), z is zero at all nodes that belong to at most two subdomains. The remaining nodes lie on crosspoints or edges (in the 3D case) of subdomains. From the definition of B, at every such node x, $z_i(x)$ is a linear combination of the entries of Au that correspond to the same node x, and the coefficients of the linear combinations are bounded only in terms of the number of subdomains to which the node belongs. Using Lemma 7 for the crosspoints of subdomains, we obtain for the 2D case that

$$\|A(I - (BA)^{-1})u\|_W^2 \leq C \sum_{x \text{ crosspoint}, x \in \partial\Omega_i} ((Au)_i(x))^2 \leq C(1 + \log(H/h)) \|Au\|_W^2. \tag{31}$$

In the 3D case, the argument for subdomain crosspoints is the same. In addition, we note that the coefficients of the linear combination do not change along a subdomain edge, so it remains to apply Lemma 7 on every edge.

Let us now turn to the special cases that give $\alpha = 0$.

If $BA = I$, we choose $\tilde{w} = Au$ in the following and get

$$\inf_{\substack{\tilde{w}\in W \\ B\tilde{w}=u}} \|\tilde{w}\|_W \leq \|Au\|_W \quad \text{as} \quad B(Au) = u,$$

which proves the special case 1.

Now we prove special case 2. From the definition of the $H^{1/2}$ norm [18] and the fact that Au is a piecewise linear function, it follows that

$$\|Au\|_W^2 \geq \sum_{i=1}^{n_s} |Au|_{1/2,\partial\Omega_i}^2 \geq \sum_{\substack{x \text{ crosspoint}, x\in\partial\Omega_i \\ y \text{ adjacent to x}, y\in\partial\Omega_i}} \left((Au)_i(x) - (Au)_i(y)\right)^2. \tag{32}$$

For any crosspoint x, it follows from the assumption that for every $\bar{w} \in$ Range A,

$$\sum_{\substack{i,\,\partial\Omega_i\ni x \\ y \text{ adjacent to } x,\ y\in\partial\Omega_i}} \left(\bar{w}_i(x) - \bar{w}_i(y)\right)^2 = 0 \Rightarrow \sum_{i,\partial\Omega_i\ni x} \left(\bar{w}_i(x)\right)^2 = 0.$$

Consequently, by compactness, and since there are only finitely many different numbers of subdomains sharing a crosspoint,

$$\sum_{i,\,\partial\Omega_i\ni x} \left(\bar{w}_i(x)\right)^2 \leq C \sum_{\substack{i,\,\partial\Omega_i\ni x \\ y \text{ adjacent to } x,\ y\in\partial\Omega_i}} \left(\bar{w}_i(x) - \bar{w}_i(y)\right)^2, \qquad \forall \bar{w} \in \text{Range } A.$$

By summation over all crosspoints x and using (31) and (32), we get

$$\|A(I - (BA)^{-1})u\|_W^2 \leq C\|Au\|_W^2,$$

which concludes the proof of this case.

In order to prove case 3, we verify the assumptions of case 2. We formulate only the proof for a crosspoint shared by three subdomains (Fig. 1). The proof is similar for a different odd number of subdomains. Let $\bar{w} \in$ Range A. Since $\bar{w}_1(x_\beta) - \bar{w}_1(x_\alpha) = 0$ and $\bar{w}_1(x_\beta) - \bar{w}_1(x_\delta) = 0$, we have $\bar{w}_1(x_\alpha) = \bar{w}_1(x_\delta)$. Similarly, we obtain $\bar{w}_2(x_\alpha) = \bar{w}_2(x_\gamma)$ and $\bar{w}_3(x_\delta) = \bar{w}_3(x_\gamma)$. Now $\bar{w} \in$ Range A implies $\bar{w}_1(x_\alpha) = -\bar{w}_2(x_\alpha)$, $\bar{w}_2(x_\gamma) = -\bar{w}_3(x_\gamma)$, and $\bar{w}_3(x_\delta) = -\bar{w}_1(x_\delta)$, which can be satisfied only if $\bar{w}_1(x_\alpha) = \bar{w}_1(x_\delta) = ... = 0$. □

Remark 9 *In general, the exponent $\alpha = 1$ in Lemma 8 cannot be improved. To see that, let us consider the configuration with values of u and Au in the neighborhood of a crosspoint as in Fig. 2; these values violate the assumptions of special case 2.*

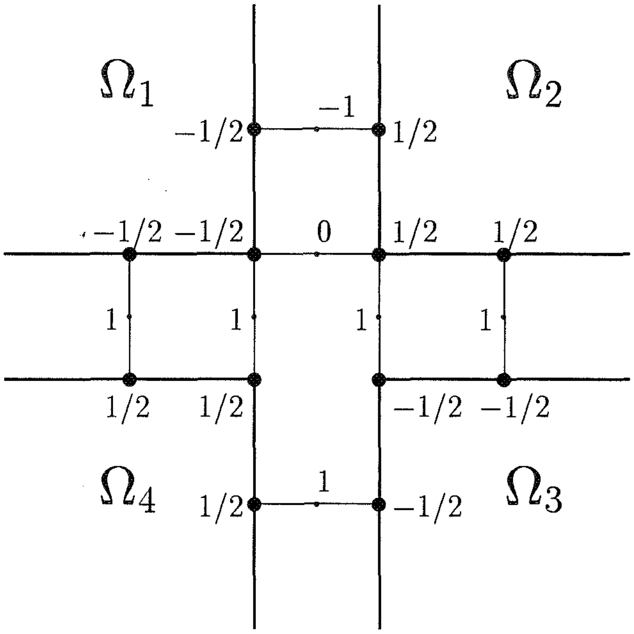


Figure 2: Counterexample.

Extending the values of u in Fig. 2 to decay as $\log^\gamma(t/H)$, $\gamma < 1/2$, where t is the distance from the crosspoint, we obtain a function $u \in U$ such that

$$\|Au\|_W \approx C, \qquad \|u\|_{H^{1/2}(\partial\Omega_1\cap\partial\Omega_2)} \approx |\log h/H|^\gamma.$$

If $u = Bw$, then on $\partial\Omega_1 \cap \partial\Omega_2$, $u = w_2 - w_1$, we obtain

$$\begin{aligned}|u|_{H^{1/2}(\partial\Omega_1\cap\partial\Omega_2)} &\le |w_1|_{H^{1/2}(\partial\Omega_1\cap\partial\Omega_2)} + |w_2|_{H^{1/2}(\partial\Omega_1\cap\partial\Omega_2)} \\ &\le |w_1|_{H^{1/2}(\partial\Omega_1)} + |w_2|_{H^{1/2}(\partial\Omega_2)} \\ &\le \|w\|_W\end{aligned}$$

so $\inf_{Bw=u}\|w\|_W \ge C(\gamma)|\log h/H|^\gamma$ *for all* $\gamma < 1/2$.

Lemma 10 *Let $\lambda \in V$. Then for all $w \in W$, there is a $\tilde{w} \in W$ such that $AB\tilde{w} \perp \operatorname{Ker} S$ and*

$$\frac{\langle\lambda, Bw\rangle^2}{\|w\|_W^2} \le C(1+\log H/h)^2\frac{\langle\lambda, B\tilde{w}\rangle^2}{\|AB\tilde{w}\|_W^2}.$$

Proof. Let $w \in W$ be arbitrary, and put $\tilde{w} = w + z$ where $z \in \operatorname{Ker} S$. Since $\lambda \in V$, we have

$$\langle\lambda, Bw\rangle = \langle\lambda, B\tilde{w}\rangle. \tag{33}$$

We would like to have $AB\tilde{w} \perp \operatorname{Ker} S$, which can be also written as

$$\langle Bz, B\tilde{z}\rangle = -\langle Bw, B\tilde{z}\rangle \quad \forall\tilde{z} \in \operatorname{Ker} S.$$

The bilinear form $\langle B\cdot, B\cdot\rangle$ is an inner product on the factorspace Ker $S/(\text{Ker } S \cap \text{Ker } B)$, so by Riesz representation theorem we may conclude that there exists $z \in$ Ker S satisfying $\|Bz\| \leq \|Bw\|$.

Now, from the definition of B and the norm in W, we obtain

$$\|Bw\|^2 \leq C\|w\|^2 \leq CH\|w\|_W^2.$$

Also, since $z \in$ Ker S, it is constant on each $\partial\Omega_i$, and we have the following by Lemma 7

$$\|ABz\|_W^2 \leq C/H\|Bz\|^2(1+\log H/h)^2.$$

Together this yields

$$\|ABz\|_W^2 \leq C(1+\log H/h)^2\|w\|_W^2.$$

By the definition of A and B, $(ABw)_i$ on $\partial\Omega_i \cup \partial\Omega_j$ is a linear combination (with bounded coefficients) of (a bounded number of) w_k from all $\partial\Omega_k$ adjacent to $\partial\Omega_i \cup \partial\Omega_j$. From Lemma 7,

$$\|ABw\|_W \leq C(1+\log(H/h))\|w\|_W, \quad \forall w \in W.$$

Finally, summarizing,

$$\|AB\tilde{w}\|_W \leq \|ABw\|_W + \|ABz\|_W \leq C(1+\log H/h)\|w\|_W.$$

From this and (33), the result follows. □

We have now everything ready to prove the estimate (21).

Lemma 11 $c(1+\log(H/h))^{-\alpha}\|\lambda\|_{V'}^2 \leq \langle\lambda, F\lambda\rangle \leq C(1+\log(H/h))^2\|\lambda\|_{V'}^2$, $\forall\lambda \in V$, *with α defined in Lemma 8.*

Proof. From the spectral decomposition (14), define $S^{-1/2} = \sum_{t>0} t^{-1/2}v_t v_t'$. Then $S^+ = S^{-1/2}S^{-1/2}$, and for $\lambda \in V$,

$$\begin{aligned}
\langle\lambda, F\lambda\rangle &= \langle S^+B'\lambda, B'\lambda\rangle = \langle S^{-1/2}B'\lambda, S^{-1/2}B'\lambda\rangle \\
&= \|S^{-1/2}B'\lambda\|^2 = \sup_{x\in W}\frac{\langle S^{-1/2}B'\lambda, x\rangle^2}{\|x\|^2} = \sup_{\substack{x\in W,\ x=x_1+x_2 \\ x_1\in\text{Ker } S\ \ x_2\perp\text{Ker } S}}\frac{\langle B'\lambda, S^{-1/2}x\rangle^2}{\|x_1+x_2\|^2} \\
&= \sup_{x_2\in W,\ x_2\perp\text{Ker } S}\frac{\langle B'\lambda, S^{-1/2}x_2\rangle^2}{\|x_2\|^2}
\end{aligned}$$

since $S^{-1/2}x_1 = 0$ and $\|x\|^2 = \|x_1\|^2 + \|x_2\|^2$. Now write any $w \in W$ as

$$w = w_1 + w_2, \qquad w_1 \in \text{Ker } S, \qquad w_2 = S^{-1/2}x_2 \perp \text{Ker } S.$$

From the definition of V in (15), $\lambda \in V$ implies that

$$\langle B'\lambda, w_1\rangle = 0.$$

Since

$$\|x_2\|^2 = \langle x_2, x_2\rangle = \langle w_2, Sw_2\rangle \approx |w_2|_W \approx \|w_2\|_W$$

from Lemma 3 and Lemma 4, it follows that

$$\langle \lambda, F\lambda\rangle = \sup_{w_2\in W,\; w_2\perp \mathrm{Ker}\; S} \frac{\langle B'\lambda, w_2\rangle^2}{\langle w_2, Sw_2\rangle} \approx \sup_{w\in W} \frac{\langle \lambda, Bw\rangle^2}{\|w\|_W^2}.$$

Lemma 8 shows that

$$\begin{aligned}\sup_{w\in W} \frac{\langle \lambda, Bw\rangle^2}{\|w\|_W^2} &= \sup_{w\in W} \frac{\langle \lambda, Bw\rangle^2}{\inf_{Bv=Bw} \|v\|_W^2} \geq \frac{1}{C(1+\log H/h)^\alpha} \sup_{w\in W} \frac{\langle \lambda, Bw\rangle^2}{\|ABw\|_W^2} \\ &\geq \frac{1}{C(1+\log H/h)^\alpha} \sup_{\substack{w\in W \\ ABw\perp \mathrm{Ker}\; S}} \frac{\langle \lambda, Bw\rangle^2}{\|ABw\|_W^2}.\end{aligned}$$

Lemma 10 yields an upper bound

$$\sup_{w\in W} \frac{\langle \lambda, Bw\rangle^2}{\|w\|_W^2} \leq C(1+\log H/h)^2 \sup_{\substack{w\in W \\ ABw\perp \mathrm{Ker}\; S}} \frac{\langle \lambda, Bw\rangle^2}{\|ABw\|_W^2}.$$

Finally, by definition of the norm $\|\cdot\|_{V'}$,

$$\sup_{\substack{w\in W \\ ABw\perp \mathrm{Ker}\; S}} \frac{\langle \lambda, Bw\rangle}{\|ABw\|_W} = \sup_{v\in V} \frac{\langle \lambda, v\rangle}{\|Av\|_W} = \|\lambda\|_{V'}$$

since B spans V. □

Condition Number Estimate

The final result now follows from the abstract estimate in Lemma 1 with the assumptions verified by Lemma 6 and Lemma 11.

Theorem 12 *The condition number of the FETI method with the Dirichlet preconditioner satisfies*

$$\kappa = \frac{\lambda_{max}(P_V M P_V F)}{\lambda_{min}(P_V M P_V F)} \leq C(1+\log\frac{H}{h})^\gamma$$

with $\gamma = 3$, and $\gamma = 2$ in the special cases listed in Lemma 8.

REFERENCES

[1] P. E. Bjørstad and J. Mandel, *Spectra of sums of orthogonal projections and applications to parallel computing*, BIT, 31 (1991), pp. 76–88.

[2] J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., 47 (1986), pp. 103–134.

[3] ——, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp., 53 (1989), pp. 1–24.

[4] Y.-H. de Roeck and P. LeTallec, *Analysis and test of a local domain decomposition*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. A. Kuznetsov, G. A. Meurant, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1991, SIAM, pp. 112–128.

[5] Q. V. Dinh, R. Glowinski, J. He, V. Kwock, T. W. Pan, and J. Périaux, *Lagrange multiplier approach to fictitious domain methods: application to fluid dynamics and electro-magnetics*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, 1992, SIAM, pp. 151–194.

[6] M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Numer. Anal., 31 (1994), pp. 1662–1694.

[7] 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.

[8] ——, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Tech. Report 626, Department of Computer Science, Courant Institute, March 1993. To appear in Comm. Pure Appl. Math.

[9] C. Farhat, P. S. Chen, and J. Mandel, *Scalable Lagrange multiplier based domain decomposition method for time-dependent problems*, Int. J. Numer. Meth. Engrg., (1995). To appear.

[10] C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Engrg., 115 (1994), pp. 367–388.

[11] 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).

[12] R. GLOWINSKI AND M. F. WHEELER, *Domain decomposition and mixed finite element methods for elliptic problems*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM.

[13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, John Hopkins University Press, second ed., 1989.

[14] P. L. LIONS, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM.

[15] J. MANDEL, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.

[16] J. MANDEL AND M. BREZINA, *Balancing domain decomposition for problems with large jumps in coefficients.* Submitted, 1993.

[17] ——, *Balancing domain decomposition: Theory and computations in two and three dimensions*, UCD/CCM Report 2, Center for Computational Mathematics, University of Colorado at Denver, 1993.

[18] J. NEČAS, *Les méthodes directes en théorie des équations elliptiques*, Academia, Prague, 1967.

[19] O. B. WIDLUND, *An extension theorem for finite element spaces with three applications*, in Numerical Techniques in Continuum Mechanics, W. Hackbusch and K. Witsch, eds., Braunschweig/Wiesbaden, 1987. Also in Notes on Numerical Fluid Mechanics, v. 16, Friedr. Vieweg und Sohn, pp. 110–122. Also in Proceedings of the Second GAMM-Seminar, Kiel, January, 1986.

A Systematic Solution Approach for Neutron Transport Problems in Diffusive Regimes *

T. A. Manteuffel[†] K. J. Ressel[‡]

SUMMARY

A systematic solution approach for the neutron transport equation, based on a least-squares finite-element discretization, is presented. This approach includes the theory for the existence and uniqueness of the analytical as well as of the discrete solution, bounds for the discretization error, and guidance for the development of an efficient multigrid solver for the resulting discrete problem. To guarantee the accuracy of the discrete solution for diffusive regimes, a scaling transformation is applied to the transport operator prior to the discretization. The key result is the proof of the V-ellipticity and continuity of the scaled least-squares bilinear form with constants that are independent of the total cross section and the absorption cross section. For a variety of least-squares finite-element discretizations this leads to error bounds that remain valid in diffusive regimes. Moreover, for problems in slab geometry a full multigrid solver is presented with $V(1,1)$-cycle convergence rates approximately equal to 0.1, independent of the size of the total cross section and the absorption cross section.

1. INTRODUCTION

The deterministic numerical solution of neutron transport problems becomes hard in diffusive regimes, which are characterized by very large total cross sections and very

*This work was supported by the DOE under grant DE-FG03-93ER25165 and the NSF under grant DMS-8704169.

[†] Program in Applied Mathematics, University of Colorado at Boulder, CB 526, Boulder, CO 80309-0526 (tmanteuf@sobolev.colorado.edu).

[‡] Interdisciplinary Project Center for Supercomputing (IPS), Clausiusstraße 59, RZ F-11, ETHZ, CH-8092 Zürich, Switzerland (kjr@ips.id.ethz.ch).

small absorption cross sections. In these regimes the transport equation is nearly singular and its solution in the interior of the computational domain is close to the solution of a diffusion equation. In order to solve diffusive transport problems numerically, it is advantageous to use a discretization for the transport operator that resembles a good approximation of a diffusion operator in diffusive regimes. In the past, special discretizations for transport problems in slab geometry have been developed that have this property. Among them are the Diamond Difference scheme (Lewis and Miller [16]), the Linear Discontinuous scheme (Alcouffe et al. [2]) and the Modified Linear Discontinuous scheme (Larsen and Morel [15]). However, these discretizations have the disadvantage that either the solution of the resulting discrete system (Manteuffel et al. [17] [18]) or their extension to higher dimensions is difficult.

In this paper we present a general framework for constructing discretizations of transport problems that are accurate in diffusive regimes. This framework, which is based on a least-squares variational formulation in combination with a scaling transformation, represents a systematic solution approach since it includes the theory for the existence and uniqueness of the analytical, as well as of the discrete, solution, bounds for the discretization error, and guidance for the development of an efficient multigrid solver for the resulting discrete problem.

To introduce our notation we recall that the single group, steady state, isotropic form of the neutron transport equation is given by (Lewis and Miller [16])

$$\left\{ \begin{array}{l} [\underline{\Omega} \cdot \underline{\nabla} + \sigma_t I - \sigma_s P]\, \psi(\underline{r}, \underline{\Omega}) = \; q(\underline{r}, \underline{\Omega}) \quad \text{for } (\underline{r}, \underline{\Omega}) \in \mathcal{R} \times S^1 \\ \psi(\underline{r}, \underline{\Omega}) = g(\underline{r}, \underline{\Omega}) \quad \text{for } \underline{r} \in \partial\mathcal{R} \;\wedge\; \underline{n}(\underline{r}) \cdot \underline{\Omega} < 0 \end{array} \right\}, \tag{1.1}$$

where σ_t is the total cross section, σ_s is the scattering cross section, and $\psi(\underline{r}, \underline{\Omega})$ is the *angular flux*, to be determined for all points $\underline{r} = (x, y, z)$ in a region $\mathcal{R} \subset I\!R^3$ with a sufficiently smooth boundary (for example of class $C^{1,1}$ (Grisvard [10, p. 5]) and all possible travel directions $\underline{\Omega}$ on the unit sphere S^1). The operator P is defined by

$$P\psi(\underline{r}, \underline{\Omega}) := \frac{1}{4\pi} \int\limits_{S^1} \psi(\underline{r}, \underline{\Omega}')\, d\Omega', \tag{1.2}$$

which is an L^2-projection onto the space of functions that are independent of direction angle $\underline{\Omega}$. The boundary conditions specify the inflow of particles into the region $\mathcal{R}$, since $\underline{n}(\underline{r})$ denotes the unit outgoing normal at $\underline{r} \in \partial\mathcal{R}$. Such problems arise as the inner loop of time-dependent, multienergy-group problems.

In the case of *slab geometry* it is assumed that $\frac{\partial \psi}{\partial x} = \frac{\partial \psi}{\partial y} \equiv 0$, so that $\psi(\underline{r}, \underline{\Omega}) = \psi(z, \mu)$ with $\mu := \cos(\theta)$, where θ denotes the angle between $\underline{\Omega}$ and the z-axis. Equation (1.1) reduces then to [16]

$$\left\{\begin{array}{rll} \left[\mu\dfrac{\partial}{\partial z}+\sigma_t I-\sigma_s P\right]\psi(z,\mu) & = q(z,\mu) & \text{for } (z,\mu)\in[z_l,z_r]\times[-1,1] \\ \psi(z_l,\mu) & = g_l(\mu) & \text{for } \mu>0 \\ \psi(z_r,\mu) & = g_r(\mu) & \text{for } \mu<0 \end{array}\right\}. \tag{1.3}$$

Now, the operator P is defined by

$$P\psi(z,\mu):=\frac{1}{2}\int_{-1}^{1}\psi(z,\mu')\,d\mu', \tag{1.4}$$

which is an L^2-projection onto the space of all functions that are independent of μ.

Without loss of generality, we assume in the following vacuum boundary conditions ($g(\underline{r},\underline{\Omega})\equiv 0$ in (1.1) and $g_l(\mu)\equiv g_r(\mu)\equiv 0$ in (1.3), respectively) and further that $\operatorname{diam}(\mathcal{R})=1$ in (1.1) and $|z_r-z_l|=1$ in (1.3), respectively. Both assumptions can be established by a simple transformation.

When $\sigma_t\to\infty$ and $\frac{\sigma_s}{\sigma_t}\to 1$, equations (1.1) and (1.3) become singular. Dividing (1.1) or (1.3) by σ_t results in the limit equation $(I-P)\psi=0$. Therefore, the limit solution is independent of direction angle $\underline{\Omega}$ and μ, respectively. Moreover, when $\sigma_t\to\infty$ and $\frac{\sigma_s}{\sigma_t}\to 1$ in a certain way, which is called the *diffusion limit*, it can be shown (Larsen [13]) that the limit solution converges to a solution of a diffusion equation. To be more specific, we introduce the absorption cross section $\sigma_a:=\sigma_t-\sigma_s$ and a small parameter ε. The diffusion limit can then be defined as the limit $\varepsilon\to 0$ after scaling the cross sections and the source in the following way:

$$q(\underline{r},\underline{\Omega})\to\varepsilon q(\underline{r},\underline{\Omega}),\quad \sigma_t\to\frac{1}{\varepsilon},\quad \sigma_a\to\varepsilon\alpha, \tag{1.5}$$

where α is assumed to be $O(1)$. In this parameterization the transport equation becomes

$$\tilde{\mathcal{L}}\psi(\underline{r},\underline{\Omega}):=\left[\underline{\Omega}\cdot\underline{\nabla}+\frac{1}{\varepsilon}(I-P)+\varepsilon\alpha P\right]\psi(\underline{r},\underline{\Omega})=\varepsilon q(\underline{r},\underline{\Omega}). \tag{1.6}$$

Using an asymptotic expansion in ε it can be proven (Larsen [13], Pomraning [24]) that the solution of (1.6) has the *diffusion expansion*

$$\psi(\underline{r},\underline{\Omega})=\phi_0(\underline{r})+\varepsilon\phi_R(\underline{r},\underline{\Omega}), \tag{1.7}$$

where ϕ_0 is, at a few mean free paths away from the boundary, a solution of the diffusion equation

$$-\nabla\cdot\frac{1}{3}\nabla\phi_0(\underline{r})+\alpha\phi_0(\underline{r})=Pq(\underline{r},\underline{\Omega}). \tag{1.8}$$

For the following analysis of a least-squares finite-element discretization of the transport equation (1.1) we use the form of the transport operator in (1.6).

This paper is organized as follows. In Section 2, we describe briefly the least-squares finite-element discretization. Further, we introduce and motivate in this section a scaling transformation that is applied to the transport operator prior to the discretization in order to ensure the accuracy of the discrete solution for diffusive regimes. In Section 3, we state that the scaled least-squares bilinear form is continuous and V-elliptic in a certain norm with constants independent of ε and α. The existence and uniqueness of the analytical, as well as of the discrete, problem then follows directly from the Lax-Milgram Lemma [7].

Furthermore, the continuity and the V-ellipticity, in combination with Céa's Lemma [7], are the basis for discretization error bounds that are established in Section 4 for a variety of conforming finite-element spaces. Since the continuity and the V-ellipticity constants are independent of ε and α, these error bounds remain valid for diffusive regimes. Thus, the least-squares discretization of the scaled transport equation with simple conforming finite-elements yields an accurate discrete solution, even in diffusive regimes. In Section 5, we describe a full multigrid solver for problems in slab geometry and present some convergence rates. Finally, in Section 6 we draw some conclusions.

2. SCALING TRANSFORMATION

Let us denote the standard inner product and associated norm of $L^2(\mathcal{R} \times S^1)$ by

$$\langle u, v\rangle := \int_{\mathcal{R}} \int_{S^1} u \cdot v^* \, d\Omega dr; \qquad \|u\| := \sqrt{\langle u, u\rangle} \quad \forall\, u, v \in L^2(\mathcal{R} \times S^1), \tag{2.1}$$

where v^* is the complex conjugate[1] of v. Further, let V be a Hilbert space with underlying norm $\|\cdot\|_V$, which we will specify later. Then, the least-squares variational formulation of (1.1) is given by (see (1.6))

$$\min_{\psi \in V} \tilde{F}(\psi), \qquad \text{with } \tilde{F}(\psi) := \int_{\mathcal{R}} \int_{S^1} \left|\tilde{\mathcal{L}}\psi(\underline{r}, \underline{\Omega}) - q(\underline{r}, \underline{\Omega})\right|^2 d\Omega dr. \tag{2.2}$$

In order for $\psi \in V$ to be a minimizer of the functional $\tilde{F}$ in (2.2), a necessary condition is that the first variation of $\tilde{F}$ must vanish at ψ for all admissible $v \in V$, which results in the following problem: find $\psi \in V$ such that

$$\tilde{a}(\psi, v) := \left\langle \tilde{\mathcal{L}}\psi, \tilde{\mathcal{L}}v \right\rangle \;=\; \left\langle q, \tilde{\mathcal{L}}v \right\rangle \quad \forall v \in V. \tag{2.3}$$

For the least-squares finite-element discretization of (2.2), the Hilbert space V is replaced by a finite dimensional subspace $V^h \subset V$. This leads to the discrete problem:

[1] We allow here complex valued functions, since we use in Section 4 the expansion of v into spherical harmonics.

find $\psi_h \in V^h$ such that

$$\widetilde{a}(\psi_h, v_h) = \left\langle q, \widetilde{\mathcal{L}} v_h \right\rangle \quad \forall v_h \in V^h. \tag{2.4}$$

By an asymptotic analysis it was shown in [19] and [25] for slab geometry and V^h formed by piecewise linear basis functions in space and a finite number of Legendre polynomials as basis functions in angle that this direct least-squares approach is not accurate in diffusive regimes. This can also be explained by the following heuristic argument. Because of the diffusion expansion (1.7) the important component of the solution ψ in diffusive regimes is the part that is independent of direction angle $\underline{\Omega}$, which is given by $P\psi$. On the other hand, the component $(I-P)\psi$ of the solution is irrelevant in diffusive regimes. By Céa's Lemma [7], the solution of the least-squares discretization can be viewed as the best approximation to the exact solution in the discrete space V^h with respect to the semi-norm $\sqrt{\widetilde{a}(\cdot,\cdot)} := \sqrt{<\widetilde{\mathcal{L}}\cdot, \widetilde{\mathcal{L}}\cdot>}$. However, the different terms in the operator $\widetilde{\mathcal{L}}$, as defined in (1.6), are unbalanced (there are $O(\frac{1}{\varepsilon})$, $O(1)$ and $O(\varepsilon)$ terms), so that different components of the approximation error are weighted differently in $\sqrt{\widetilde{a}(\cdot,\cdot)}$. The leading term of $\widetilde{\mathcal{L}}$ is $\frac{1}{\varepsilon}(I-P)$, which means that the part of the error that is dependent on angle is weighted in this norm very strongly in diffusive regimes (very small ε), even though this part is irrelevant. On the other hand, the part of the error that is independent of angle, which is the important part in diffusive regimes, is hardly measured in the semi-norm $\sqrt{\widetilde{a}(\cdot,\cdot)}$, since it is weighted by ε.

The idea is to scale equation (1.6), thus changing the weighting in the norm used in the least-squares discretization, which, in turn, alters the choice of the element of the discrtete space as an approximation to the exact solution. Let us define the following *scaling transformation* and its inverse:

$$S := P + \varepsilon(I-P), \qquad S^{-1} = P + \frac{1}{\varepsilon}(I-P). \tag{2.5}$$

Clearly, applying the scaling transformation S from the left to the transport equation prior to the least-squares discretization will increase the weight of the important error component and decrease the weight for the irrelevant component. After applying the scaling transformation S from the left and dividing by ε, equation (1.6) becomes

$$\mathcal{L}\psi := \frac{1}{\varepsilon} S\widetilde{\mathcal{L}}\psi = \frac{1}{\varepsilon} S\underline{\Omega} \cdot \underline{\nabla}\psi + \frac{1}{\varepsilon}(I-P)\psi + \alpha P\psi = q_s, \tag{2.6}$$

with $q_s := Sq$.

Equation (2.6) can be balanced further by applying the scaling transformation S also from the right. Let the domain of operator $\mathcal{L}$ in (2.6) be the Hilbert space V. Then, we define a space $\widehat{V}$ by

$$\widehat{V} := S^{-1}V, \tag{2.7}$$

so that

$$\widehat{v} = S^{-1}v \text{ and } S\widehat{v} = v \tag{2.8}$$

for all $v \in V$ and $\widehat{v} \in \widehat{V}$. Scaling (2.6) from the right results in

$$\mathcal{L}SS^{-1}\psi = \mathcal{L}S\widehat{\psi} = \underline{Q} \cdot \underline{\nabla}\widehat{\psi} + (I - P)\widehat{\psi} + \alpha P\widehat{\psi} = q_s, \tag{2.9}$$

where

$$\underline{Q} := \frac{1}{\varepsilon} S\underline{\Omega}S = (1 - \varepsilon)\left(P\underline{\Omega} + \underline{\Omega}P\right) + \varepsilon\underline{\Omega}I.$$

In the double-scaled operator $\mathcal{L}S$ in (2.9) the derivative of zeroth moment $(P\underline{\nabla}\psi)$, the derivative of the first moments $(P\underline{\Omega} \cdot \underline{\nabla}\psi)$ and all components of ψ themselves are weighted equally. Moreover, it is easily seen that the double-scaled operator $\mathcal{L}S$ goes to a bounded nonsingular limit operator as $\varepsilon \to 0$.

In the least-squares context, the additional scaling from the right can be avoided because

$$\min_{\widehat{\psi} \in \widehat{V}} \left\langle \mathcal{L}S\widehat{\psi} - q_s, \mathcal{L}S\widehat{\psi} - q_s \right\rangle \iff \min_{\psi \in V} \left\langle \mathcal{L}\psi - q_s, \mathcal{L}\psi - q_s \right\rangle, \tag{2.10}$$

which will simplify the boundary conditions and so the computations. However, for the theory we exploit the nice form of the double-scaled operator $\mathcal{L}S$ and use this form of the transport operator as a tool.

The least-squares variational formulation of the single-scaled equation (2.6) is given by the problem: find $\psi \in V$ such that

$$a(\psi, v) := \langle \mathcal{L}\psi, \mathcal{L}v \rangle = \langle q_s, \mathcal{L}v \rangle \quad \forall v \in V. \tag{2.11}$$

For the sake of completeness we remark that for slab geometry the form of the scaling transformation S, as defined in (2.5), remains the same, except that for P the definition (1.4) has to be used. In the case of slab geometry, therefore, equation (2.6) reduces to

$$\mathcal{L}\psi := \frac{1}{\varepsilon} S\tilde{\mathcal{L}}\psi = \frac{1}{\varepsilon} S\mu\frac{\partial \psi}{\partial z} + \frac{1}{\varepsilon}\left(I - P\right)\psi + \alpha P\psi = q_s. \tag{2.12}$$

3. CONTINUITY AND V-ELLIPTICITY

In this section we summarize without proof that the scaled least-squares bilinear form (2.11) is *continuous* , i.e., there exists a constant $C_c > 0$ such that for every $u, v \in V$

$$|a(u, v)| \leq \; C_c \, \|u\|_V \, \|v\|_V, \tag{3.1}$$

and *V-elliptic* , i.e., there exists a constant $C_e > 0$ such that for all $v \in V$:

$$a(v,v) \geq \; C_e \left\|v\right\|_V^2. \tag{3.2}$$

The Hilbert space V and its norm $\|\cdot\|_V$ are specified below. It is crucial to prove these bounds with constants C_e and C_c that are independent of ε and α, since this makes it possible to establish discretization error bounds that remain valid in diffusive regimes.

We first consider the slab geometry case. Let $D := [z_l, z_r] \times [-1, 1]$ denote the computational domain and let $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the standard inner product and the associated norm of $L^2(D)$, which are defined by

$$\langle u, v \rangle := \int\limits_{x_l}^{x_r} \int\limits_{-1}^{1} u \cdot v \; d\mu dx \quad \text{and} \quad \|u\| := \sqrt{\langle u, u \rangle}.$$

An appropriate norm for bounding the least-squares bilinear form $a(\cdot, \cdot)$ is then given by the norm

$$\|v\|_V^2 := \left\| \frac{1}{\varepsilon} S \mu \frac{\partial v}{\partial z} \right\|^2 + \left\| \frac{1}{\varepsilon} (I - P) v \right\|^2 + \|Pv\|^2. \tag{3.3}$$

The Hilbert space V can then be defined by

$$V := \overline{\left\{ v \in C^\infty(\overline{D});\; v(z_l, \mu) = 0 \text{ for } \mu > 0;\; v(z_r, \mu) = 0 \text{ for } \mu < 0 \right\}}, \tag{3.4}$$

where the closure is taken with respect to the norm $\| \cdot \|_V$.

From the Cauchy-Schwarz inequality and discrete Hölder inequality it is easy to obtain that for all $u, v \in V$

$$|a(u,v)| = |\langle \mathcal{L}u, \mathcal{L}v \rangle| \leq \; \|\mathcal{L}u\| \; \|\mathcal{L}v\| \leq 3 \|u\|_V \; \|v\|_V. \tag{3.5}$$

Thus, the bilinear form (2.11) is continuous with respect to the norm $\|\cdot\|_V$ with $C_c = 3$.

The proof of the V-ellipticity is much harder and requires several technical lemmas. For a proof of the following theorem we refer the reader to [20] and [25].

Theorem 3.1 (V-ellipticity of $a(\cdot, \cdot)$) *Suppose that $0 \leq \alpha \leq 1$, $0 \leq \varepsilon < \frac{1}{\sqrt{3}}$. Let $a(\cdot, \cdot)$ and $\|\cdot\|_V$ be given as in (2.11) and (3.3) respectively. Then, there exists a constant $C_e > 0$ such that for all $v \in V$,*

$$a(v,v) \; \geq \; C_e \left\|v\right\|_V^2, \tag{3.6}$$

where $C_e = 0.012$, which is independent of ε and α. □

In the case of x-y-z-geometry we let $D := \mathcal{R} \times S^1$ and generalize the definition of $\|\cdot\|_V$ in (3.3) and the Hilbert space V in the following way:

$$\|v\|_V^2 := \left\|\frac{1}{\varepsilon} S \underline{\Omega} \cdot \underline{\nabla} v\right\|^2 + \left\|\frac{1}{\varepsilon}(I-P)v\right\|^2 + \|Pv\|^2 . \tag{3.7}$$

$$V := \overline{\left\{v \in C^\infty(\overline{D});\ v(\underline{r}, \underline{\Omega}) = 0 \text{ for } \underline{r} \in \partial\mathcal{R}, \text{ and } \underline{\Omega} \cdot \underline{n}(\underline{r}) < 0\right\}}, \tag{3.8}$$

where the closure is now taken with respect to the norm $\|\cdot\|_V$ in (3.7) and $\|\cdot\|$ denotes the norm in (2.1). The continuity (3.5) and the V-ellipticity (3.6) hold then with exactly the same constants C_c and C_e as in the slab geometry case.

Together with the Lax-Milgram Lemma [7] the existence and the uniqueness of a solution for problem (2.11) and its discrete version (4.1), where V is replaced by a finite dimensional subspace $V^h \subset V$, follows directly. In the next section we will use the continuity and the V-ellipticity of the bilinear form $a(\cdot,\cdot)$ to prove discretization error bounds for a variety of discrete spaces V^h.

4. DISCRETIZATION ERROR BOUNDS

In this section we establish bounds for the discretization error $\psi - \psi_h$. Here, $\psi \in V$ denotes the solution of (2.11) and $\psi_h \in V^h \subset V$ denotes the solution of the corresponding discrete problem: find $\psi_h \in V^h$ such that

$$a(\psi_h, v_h) = \langle q_s, \mathcal{L} v_h\rangle \qquad \forall v_h \in V^h. \tag{4.1}$$

The continuity and the V-ellipticity of $a(\cdot,\cdot)$ lead directly to Céa's Lemma [7]:

$$a(\psi - \psi_h, \psi - \psi_h) \leq a(\psi - v_h, \psi - v_h) \qquad \forall v_h \in V^h \tag{4.2}$$

or

$$\|\psi - \psi_h\|_V \leq \sqrt{\frac{C_c}{C_e}} \min_{v_h \in V^h} \|\psi - v_h\|_V . \tag{4.3}$$

Therefore, bounding $\|\psi - \psi_h\|_V$ is reduced to the problem of bounding $\min\limits_{v_h \in V^h} \|\psi - v_h\|_V$, which is a problem of approximation theory and depends on the space V^h. Here we consider discrete spaces V^h that are formed by functions that can be expanded into the first N Legendre polynomials (spherical harmonics in the case of x-y-z-geometry) with respect to the direction angle μ ($\underline{\Omega}$) and are piecewise polynomials of degree k in z ($\underline{r}$) on a partition $\mathcal{T}_h$ of the slab $[z_l, z_r]$ (region $\mathcal{R}$). This class of finite dimensional subspaces corresponds to a discretization by a spectral method in angle and a finite-element discretization in space. The spectral discretization in angle with Legendre polynomials (spherical harmonics) is common for transport problems [16] and also called a P_N-discretization.

Again, we consider the slab geometry case first. Let $\mathcal{T}_h = \{z_l =: z_0, z_1, \ldots, z_m := z_r\}$ be a partition of the slab $[z_l, z_r]$ with maximum mesh size h and let $I\!P_k(\mathcal{T}_h)$ denote the space of piecewise polynomials of degree $\leq k$ on the partition $\mathcal{T}_h$. Further, let $P_l(\mu)$ denote the l-th Legendre polynomial. The normalized Legendre polynomials $p_l(\mu) := \sqrt{2l+1}\,P_l(\mu)$ form then an orthonormal basis of $L^2([-1,1])$. Thus, any $\psi \in V$ has the following expansion in angle,

$$\psi(z,\mu) = \sum_{l=0}^{\infty} \phi_l(z)\, p_l(\mu), \tag{4.4}$$

where the Fourier coefficients $\phi_l(z)$, which are called *moments* in transport theory, are given by

$$\phi_l(z) = \frac{1}{2}\int_{-1}^{1} \psi(z,\mu)\, p_l(\mu) d\mu. \tag{4.5}$$

For the discretization we truncate the expansion in (4.4) and approximate the moments $\phi_l(z)$ by piecewise polynomials on the partition $\mathcal{T}_h$. This results in the discrete space

$$\begin{aligned} V^h \; := \; & \Big\{ v_h \in C^0(D);\; v_h = \sum_{l=0}^{N-1} \phi_l^h(z) p_l(\mu);\;\; \phi_l^h(z) \in I\!P_r(\mathcal{T}_h) \;\text{ for }\; l = 0,\ldots,N-1; \\ & \qquad v_h(z_l,\mu) = 0 \;\text{ for }\; \mu > 0, \quad v_h(z_r,\mu) = 0 \text{ for } \mu < 0 \Big\}. \end{aligned} \tag{4.6}$$

Let $|\cdot|_{\nu,0}$ denote the standard semi norm of $H^\nu([z_l,z_r]) \times L^2([-1,1])$. Combining Céa's Lemma, standard finite-element approximation bounds and using the fact that the Legendre Polynomials are eigenfunctions of the Sturm-Liouville operator [9, p.21], that is,

$$\mathcal{L}_S p_l(\mu) := \frac{d}{d\mu}\left[\left(1-\mu^2\right)\frac{dp_l(\mu)}{d\mu}\right] = l(l+1)p_l(\mu),$$

the following discretization error bound can be established (see [20] and [25]).

Theorem 4.1 (Discretization Error bound for slab geometry) *Suppose* $0 \leq \alpha \leq 1$ *and* $0 \leq \varepsilon \leq \frac{1}{\sqrt{3}}$. *Let* $\psi \in V \cap \left(H^{k+1}([z_l,z_r]) \times H^2([-1,1])\right)$ *be the solution of* (2.11) *with* $q_s \in H^k([z_l,z_r]) \times H^2([-1,1])$. *Further, let* $\psi_h \in V^h$ *be the solution of* (4.1) *with* V^h *defined as in* (4.6). *Assume that* ψ *has the diffusion expansion* $\psi(z,\mu) = \phi_0(z) + \varepsilon\phi_R(z,\mu)$. *Then,*

$$\begin{aligned} \|\psi - \psi_h\|_V \leq \; & \frac{1}{\sqrt{C_e}}\left(\frac{C_1}{N}\|\mathcal{L}_S q_s\| + \frac{C_2}{N^2}\left\|\mathcal{L}_S \frac{\partial\psi}{\partial z}\right\|\right) \\ & + \sqrt{\frac{C_c}{C_e}}\left\{C_3 h^k\left(|\phi_0|_{k+1,0} + |\phi_R|_{k+1,0}\right) + e_h^B\right\} =: e_h, \end{aligned} \tag{4.7}$$

with C_1, C_2, C_3 independent of α and ε. In particular

$$\left\| P\left(\mu\frac{\partial(\psi-\psi_h)}{\partial z}\right)\right\| \le \varepsilon e_h, \qquad \|(I-P)(\psi-\psi_h)\| \le \varepsilon e_h,$$

$$\left\|(I-P)\left(\mu\frac{\partial(\psi-\psi_h)}{\partial z}\right)\right\| \le e_h, \qquad \|P(\psi-\psi_h)\| \le e_h. \qquad \square$$

For the definition of the boundary error e_h^B we refer to [20]. However the following remark explains the source of this error.

Remark 4.2 (Treatment of Boundary Conditions) In order to have $V^h \subset V$, which is necessary for Céa's Lemma, we incorporated the boundary conditions in the definition (4.6) of the discrete space V^h. However, in conjunction with a P_N discretization in angle, these boundary conditions can only be fulfilled by a discrete function if $\phi_l^h(z_l) = \phi_l^h(z_r) = 0$ for $l = 0, 1, \dots, N-1$. Therefore, the boundary conditions for the discrete problem are really given by $v_h(z_l,\mu) = v_h(z_r,\mu) = 0$ for $\mu \in [-1,1]$. The difference to the real boundary conditions ($v(z_l,\mu) = 0$ for $\mu \in [0,1]$ and $v(z_r,\mu) = 0$ for $\mu \in [-1,0]$) is measured in the error bound (4.7) by the term e_h^B. In diffusive regimes, where the analytical solution is nearly independent of μ, we have that $v(z_l,\mu) \approx 0$ for $\mu \in [-1,0]$ and $v(z_r,\mu) \approx 0$ for $\mu \in [0,1]$, so that e_h^B will be small. However, for nondiffusive problems, it is not, in general, true that the inflow of particles is nearly equal to the outflow. In this case, e_h^B will, in general, be large.

One way to avoid this difficulty would be to use nonconforming finite element subspaces, that is, to require that functions in the discrete subspace obey Mark or Marshak boundary conditions [8]. Since then $V^h \not\subset V$, Strang's Lemma [6] instead of Céa's Lemma must be used in order to establish error bounds.

Another, more natural, way to address this issue would be to incorporate the boundary conditions directly into the least-squares functional. For example, one could add to the bilinear form $a(\cdot,\cdot)$ in (2.11) the boundary form

$$b(\psi, v) := \varepsilon\left(\int_0^1 \mu\psi(z_l,\mu)v(z_l,\mu)d\mu - \int_{-1}^0 \mu\psi(z_r,\mu)v(z_r,\mu)\,d\mu\right)$$

and use a discrete space with functions that are free of any boundary constraint. Error bounds based on this approach will appear in a forthcoming paper. □

Remark 4.3 (Nondiffusive regimes) In order to get an error bound in (4.7) with a constant that is independent of parameter ε it is assumed in Theorem 4.1 that the analytical solution has a diffusion expansion. For regimes, where the diffusion expansion is not valid, $\frac{1}{\varepsilon}$ is of moderate size, so that there is no need for an error bound that is independent of ε. In this case the second term on the right hand side of (4.7) simplifies to

$$\sqrt{\frac{C_c}{C_e}}\left\{C_3 h^k \left(1+\frac{1}{\varepsilon}\right)|\psi|_{k+1,0}+e_h^B\right\}.$$

However, we point out that this bound will blow up in diffusive regimes, where $\frac{1}{\varepsilon}$ becomes very large. □

Now, we generalize the error bounds for slab geometry to x-y-z geometry. Let $\mathcal{T}_h$ be a triangulation of $\mathcal{R}$ into thetrahedrons of maximum diameter h. Recall that the *spherical harmonics* [3, p. 571] are defined by

$$Y_l^m(\theta,\varphi) := (-1)^m\, C_{l,m}\, P_l^m(\cos(\theta))\, e^{im\varphi},$$

for $l \geq 0$ and $-l \leq m \leq l$, where

$$C_{l,m} := \sqrt{\frac{(2l+1)(l-m)!}{(l+m)!}},$$

P_l^m denotes the associated Legendre polynomials, and θ denotes the polar angle with respect to the z-axis, while φ denotes the azimuthal angle about the z-axis. The spherical harmonics form an orthonormal basis of $L^2(S^1)$. Therefore, any $v \in L^2(\mathcal{R}\times S^1)$ has an expansion of the form

$$v(\underline{r},\underline{\Omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \phi_{l,m}(\underline{r})Y_l^m(\underline{\Omega}), \quad \text{with } \phi_{l,m}(\underline{r}) = \int_{S^1} v(\underline{r},\underline{\Omega})Y_l^{m*}(\underline{\Omega})\,d\Omega. \tag{4.8}$$

Similar to the slab geometry case, we truncate this expansion for the discretization and approximate the moments $\phi_{l,m}$ by a function $\phi_{l,m}^h \in I\!P_k(\mathcal{T}_h)$, where $I\!P_k(\mathcal{T}_h)$ denotes the space of piecewise polynomials of degree $\leq k$ on the triangulation $\mathcal{T}_h$. Thus, we define the following class of discrete spaces:

$$V^h := \left\{ v_h \in V : \; v_h(\underline{r},\underline{\Omega}) = \sum_{l=0}^{N-1}\sum_{m=-l}^{l} \phi_{l,m}^h(\underline{r})\, Y_l^m(\underline{\Omega}); \phi_{l,m}^h(\underline{r}) \in I\!P_k(\mathcal{T}_h) \right\}, \tag{4.9}$$

which correspond to a finite-element discretization in space and a P_N discretization [16] in angle.

Let $|\cdot|_{k+1,0}$ denote the semi norm of $H^{k+1}(\mathcal{R}) \times L^2(S^1)$. As in the slab geometry case, we combine Céa's Lemma, standard finite-element approximation bounds and use the fact that the spherical harmonics are the eigenfunctions of the Laplacian operator Δ_Ω on the unit sphere to obtain the following discretization error bound (see [20] and [25]).

Theorem 4.4 (Discretization Error Bound for x-y-z geometry) *Suppose $0 \leq \alpha \leq 1$ and $0 \leq \varepsilon \leq \frac{1}{\sqrt{3}}$. Let $\psi \in V \cap \left(H^{k+1}(\mathcal{R}) \times H^2(S^1)\right)$ be the solution of* (2.11) *with $q_s \in H^k(\mathcal{R}) \times H^2(S^1)$. Further, let $\psi_h \in V^h$ be the solution of* (4.1) *with V^h defined as in* (4.9). *Assume that ψ has the diffusion expansion* (1.7). *Then, we have:*

$$\begin{aligned} \|\psi - \psi_h\|_V \leq \; & \frac{1}{\sqrt{C_e}} \frac{C_1}{N} \left(\|\Delta_\Omega q_s\| + |\Delta_\Omega \psi|_{1,0}\right) \\ & + \sqrt{\frac{C_c}{C_e}} \left\{C_2 h^k \left(|\phi_0|_{k+1,0} + |\phi_R|_{k+1,0}\right) + e_h^B\right\}, \end{aligned} \tag{4.10}$$

with C_1 and C_2 independent of ε and α. □

5. MULTIGRID SOLVER

The accuracy of the least-squares discretization in combination with the scaling transformation for diffusive transport problems has been demonstrated numerically in [19], [25] and in [20]. In this section we restrict the presentation of numerical results to a full multigrid solver for problems in slab geometry. We refer the reader, who is not familiar with multigrid methods to (Briggs [5]) for an introduction and to (Hackbusch [11]) and (McCormick [21] [22] [23]) for more advanced topics.

The proper choice of the components, namely, the inter-grid transfer operators, coarse grid problems, and relaxation schemes, is essential for the efficiency of a multigrid solver. The choice of the first two components is naturally given by the least-squares variational formulation. The sequence of discrete spaces $V_1 \subset V_2 \subset \cdots \subset V_l = V^h$ determines the coarse grid problems since they are just the restriction of the variational problem to these discrete subspaces. The prolongation operator, which is a mapping from a coarse grid to the next finer grid in the grid sequence, is formed directly by composing the isomorphisms between the discrete spaces and their corresponding coordinate spaces with the injection mapping between V_{k-1} and V_k (Bramble [4]), (McCormick [23]). The restriction operators, which are mappings from a finer grid to the next coarser grid, are just the adjoints of the prolongation operators. Therefore, the only multigrid components that need to be chosen here are the sequence of discrete spaces and the relaxation.

For the discrete subspaces, we use finite-element spaces with linear basis elements on increasingly finer partitions (halving the spatial cells) of the slab.

As relaxation we employ a line moment relaxation that updates all moments simultaneously for a given spatial point. Our computational tests showed essentially no differences in the error reduction and smoothing properties of this line relaxation

Table 5.1: Multigrid convergence factors.

$V(1,1)$-cycle					
σ_t	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 0.25$	$\alpha = 0.1$	$\alpha = 0.0$
10^0	0.052	0.086	0.083	0.118	0.169
10^1	0.091	0.092	0.091	0.117	0.136
10^2	0.056	0.056	0.071	0.106	0.131
10^3	0.092	0.093	0.092	0.105	0.127
10^4	0.095	0.094	0.094	0.106	0.129
10^5	0.095	0.094	0.093	0.107	0.130
10^6	0.095	0.092	0.092	0.107	0.130
10^7	0.095	0.092	0.092	0.107	0.130
10^8	0.095	0.092	0.092	0.107	0.130
10^9	0.095	0.094	0.092	0.107	0.130
10^{10}	0.095	0.094	0.092	0.106	0.130

scheme for various different orderings of the spatial points. To save computation, we use this line relaxation scheme in a red-black fashion, since then the residual after one relaxation sweep is zero at the black points and need not be computed for the restriction to the next coarser grid. This scheme is also more amenable to advanced computer architectures.

The convergence rates for a $V(1,1)$-cycle of this multigrid algorithm, which uses one relaxation before and one relaxation after the coarse grid correction, are listed in Table 5.1. Even for values of $\sigma_t = 1/\varepsilon \geq 10^6$, we get $V(1,1)$-cycle convergence factors of order 0.1. These convergence factors are sufficient to get a solution with an error on the order of the discretization error by one single full-multigrid cycle.

6. CONCLUSION

The least-squares finite-element discretization with piecewise linear basis functions in space directly applied to the neutron transport equation does not yield a correct discrete solution in diffusive regimes. However, in combination with a scaling transformation applied to the transport operator prior to the discretization, the least-squares discretization is accurate for diffusive regimes and represents a systematic, general, solution approach.

This approach, which converts the first order transport problem into a variational form with a symmetric bilinear form, is systematic because it includes the theory for the existence and uniqueness of the analytical as well as for the discrete solution, bounds for the discretization error and guidance for the development of an efficient multigrid solver for the resulting discrete system.

The key results are the V-ellipticity and the continuity of the scaled least-squares bilinear form with constants independent of ε and α. They make it possible to establish error bounds that remain valid in diffusive regimes. Together with the freedom to choose a discrete space, this approach yields a general framework for finding discretizations for the transport equation that are accurate in diffusive regimes.

Because of its generality, this approach opens many possibilities for future work. The use of different discrete spaces can be explored. For example, one may consider finite-elements as basis functions for discretization of the angle dependence instead of Legendre polynomials or Spherical Harmonics. The boundary conditions could be incorporated directly into the least-squares functional, which would be a more appropriate treatment of the boundary conditions. Adaptive refinement could be combined with the multigrid solver in order to resolve boundary layers. Finally, it appears that it is possible to generalize the scaling transformation to anisotropic transport problems.

REFERENCES

[1] R.A. Adams, *Sobolev Spaces*, Academic Press, 1975.

[2] R.E. Alcouffe, E.W. Larsen, W.F. Miller and B.R. Wienke, *Computational Efficiency of Numerical Methods for the Multigroup, Discrete Ordinates Neutron Transport Equations: The Slab Geometry Case*, Nuclear Science and Engineering 71, pp. 111-127, 1979.

[3] G.B. Arfken, *Mathematical Methods for Physicists*, second edition, Academy Press, New York, 1971.

[4] J.H. Bramble, *Multigrid Methods*, Pitman Research Notes in Mathematics Series 294, Longman Scientific and Technical, Essex, 1993.

[5] W.L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[6] S.C. Brenner, L.R. Scott, *The Mathematical Theory of Finite Element Methods*, Texts in applied mathematics, Springer Verlag Inc., New York, 1994.

[7] P.G. Ciarlet and J.L. Lions, *Handbook of Numerical Analysis, v. II, Finite Element Methods*, Elsevier Science Publishers B. V. North-Holland, Amsterdam, 1991.

[8] J.J. DUDERSTADT AND W.R MARTIN, *Transport Theory*, John Wiley & Sons, New York, 1978.

[9] D. GOTTLIEB AND S.A. ORSZAG, *Numerical Analysis of Spectral Methods: Theory and Applications*, Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, 1977.

[10] P. GRISVARD, *Elliptic Problems in Nonsmooth Domains*, Pitman Advanced Publishing Program, Boston, 1985.

[11] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.

[12] C. JOHNSON, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1990.

[13] E.W. LARSEN, *Diffusion Theory as an Asymptotic Limit of Transport Theory for Nearly Critical Systems with Small Mean Free Path*, Annals of Nuclear Energy, Vol. 7, pp. 249-255.

[14] E.W. LARSEN, J.E. MOREL, AND W.F. MILLER, *Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes*, J. Comp. Phys., 69, pp. 283-324, 1987.

[15] E.W. LARSEN AND J.E. MOREL, *Asymptotic Solutions of Numerical Transport Problems in Optically Thick Diffusive Regimes II*, J. Comp. Phys. 83, (1989), p. 212.

[16] E.E. LEWIS AND W.F. MILLER, *Computational Methods of Neutron Transport*, John Wiley & Sons, New York, 1984.

[17] T.A. MANTEUFFEL, S.F. MCCORMICK, J.E. MOREL, S. OLIVEIRA AND G. YANG, *A Fast Multigrid Solver for Isotropic Transport Problems*, submitted to SIAM J. Sci. Comp., to appear.

[18] T.A MANTEUFFEL, S.F. MCCORMICK, J.E. MOREL, S. OLIVEIRA AND G. YANG, *A parallel Version of a Multigrid Algorithm for Isotropic Transport Equations*, submitted to SIAM J. Sci. and Stat. Comp. 15, No 2, pp. 474-493, March 1994.

[19] T.A. MANTEUFFEL AND K.J. RESSEL, *Multilevel Methods for Transport Equations in Diffusive Regimes*, Proceedings of the Copper Mountain Conference on Multigrid Methods, April 5-9, 1993.

[20] T.A. MANTEUFFEL AND K.J. RESSEL, *Least-Squares Finite-Element Solution of the Transport Equations in Diffusive Regimes*, in preperation.

[21] S.F. McCormick, *Multigrid Methods*, Frontiers in Applied Mathematics 3, SIAM, Philadelphia, 1987.

[22] S.F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, Frontiers in Applied Mathematics, SIAM, Philadelphia, 1989.

[23] S.F. McCormick, *Multilevel Projection Methods for Partial Differential Equations*, SIAM, Philadelphia, 1992.

[24] G.C. Pomraning, *Diffusive Limits for Linear Transport Equations*, Nuclear Science and Engineering 112, pp. 239-255, 1992.

[25] K.J. Ressel, *Least-Squares Finite-Element Solution of the Neutron Transport Equation in Diffusive Regimes*, Ph.D. thesis, University of Colorado at Denver, December 1994.

First-Order System Least-Squares for Second-Order Elliptic Problems with Discontinuous Coefficients

Thomas A. Manteuffel Stephen F. McCormick Gerhard Starke*

Abstract

The first-order system least-squares methodology represents an alternative to standard mixed finite element methods. Among its advantages is the fact that the finite element spaces approximating the pressure and flux variables are not restricted by the inf-sup condition and that the least-squares functional itself serves as an appropiate error measure. This paper studies the first-order system least-squares approach for scalar second-order elliptic boundary value problems with discontinuous coefficients. Ellipticity of an appropriately scaled least-squares bilinear form is shown independently of the size of the jumps in the coefficients leading to adequate finite element approximation results. The occurrence of singularities at interface corners and cross-points is discussed, and a weighted least-squares functional is introduced to handle such cases. Numerical experiments are presented for two test problems to illustrate the performance of this approach.

Introduction

The purpose of this paper is to apply the first-order system least-squares approach developed in [4] and [5] to scalar second-order elliptic boundary value problems in two dimensions with discontinuous coefficients. Such problems arise in various application areas, including flow in heterogeneous porous media (see, e.g., [12]), neutron transport [1], and biophysics [7]. In many physical applications, one is interested not only in an accurate approximation of the physical quantity that satisfies the scalar equation, but also in certain of its derivatives. For example, fluid flow in a porous medium can be modelled by the equation

$$-\nabla \cdot (a \nabla p) = f \tag{1}$$

for the pressure p, where the scalar function a may have large jump discontinuities across interfaces. Of particular interest here is accurate approximation of the fluid velocity

$$\mathbf{u} = a \nabla p\,, \tag{2}$$

a concern which led to the development of mixed finite element methods (see, e.g., [3, Chapter 10]). In mixed methods, both p and $\mathbf{u}$ are approximated by not necessarily identical finite elements and, roughly speaking, a Galerkin condition is imposed on the first-order system resulting from (1) and (2).

An alternative to mixed finite elements is the first-order system least-squares approach developed and analyzed, e.g., in [4], [5], [11], and [10]. This methodology replaces the Galerkin condition by the minimization of a least-squares functional associated with a first-order system derived from (1) and (2). Augmenting the basic system

*Program in Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. E-mail {tmanteuf,stevem,starke}@boulder.colorado.edu

with the curl-condition $\nabla \times (\mathbf{u}/a) = 0$ (see [5], [10]) leads to ellipticity with respect to the $H^1(\Omega)$ norm in the individual variables. Important practical advantages of this least-squares approach over standard mixed methods are: (i) the finite element spaces approximating the pressure and flux variables are not restricted by the inf-sup condition of Ladyzhenskaya-Babuška-Brezzi (cf. [3, Section 10.5]) and (ii) the least-squares functional serves as an appropriate error measure. Moreover, if the problem is sufficiently regular (e.g., if $a \in C^{1,1}(\Omega)$ and Ω has certain properties (cf. [5])), then (iii) optimal accuracy is guaranteed in each variable, including the velocities, in the H^1 norm and (iv) optimal computational complexity for the solution of the resulting discrete systems is achieved with standard multigrid methods (see [5]).

For problems with discontinuous coefficients, which is our focus in this paper, the velocity components will, in general, not be in $H^1(\Omega)$. While the theory developed in [4] and [5] already allows for discontinuous coefficients, special care must be taken in order to prove ellipticity, in an appropriate norm, with constants independent of the size of the jumps. For this purpose, an appropriate scaling of the least-squares functional that depends on the size of a in different parts of the domain is introduced. This results in ellipticity, independently of the size of coefficient jumps, and consequently in finite element approximation results, with respect to a norm that is suitably scaled depending on the size of a. This scaling is presented in the following section.

At interface corners and cross-points (i.e., where two smooth interface components intersect), the components of $\mathbf{u}$ will, in general, be unbounded, and singularities naturally arise (see, for example, Strang and Fix [14, Ch. 8]). The shape of these singularities is determined by the angle at an interface corner (or between two intersecting interfaces) and the jumps in the coefficients. We will show how the parameters describing these singularities can be computed from the coefficient jumps and corner angles. We are particularly interested in the exponent associated with the singular function at a corner or cross-points since this indicates how much we have to unweight the least-squares functional in the neighborhood of such a point. The performance of this scaled least-squares approach will be studied using bilinear finite elements for the pressure and fluxes (based on the same grid) and a full multigrid algorithm for the solution of the resulting discrete system. Finally, computational experiments for two test problems are presented.

Our restriction to two-dimensional problems is mainly for the purpose of exposition. However, some technical complications arise for three-dimensional problems. For example, two different types of singularities, associated with edges and with corners or cross-points, arise in three dimensions. We do not examine this in the present paper.

The Least-Squares Functional

Consider the following prototype problem on a bounded domain $\Omega \subset \mathbb{R}^2$:

$$\begin{aligned} -\nabla \cdot (a \nabla p) &= f, && \text{in } \Omega, \\ p &= 0, && \text{on } \Gamma_D, \\ \mathbf{n} \cdot \nabla p &= 0, && \text{on } \Gamma_N, \end{aligned} \tag{3}$$

where $\mathbf{n}$ denotes the outward unit vector normal to the boundary, $f \in L^2(\Omega)$, and $a(x_1, x_2)$ is a scalar function that is uniformly positive and bounded in Ω but may have large jumps across interfaces. We assume that $\Gamma_D \neq \emptyset$, so that the Poincaré-Friedrichs inequality

$$||p||_{0,\Omega} \leq \gamma ||\nabla p||_{0,\Omega} \tag{4}$$

holds and (3) has a unique solution in $H^1(\Omega)$. Following [5], we rewrite (3) as a first-order system by introducing the flux variable $\mathbf{u} = a\nabla p$:

$$\left\{ \begin{aligned} \mathbf{u} - a\nabla p &= 0, && \text{in } \Omega, \\ -\nabla \cdot \mathbf{u} &= f, && \text{in } \Omega, \\ p &= 0, && \text{on } \Gamma_D, \\ \mathbf{n} \cdot \mathbf{u} &= 0, && \text{on } \Gamma_N. \end{aligned} \right. \tag{5}$$

Since $\mathbf{u}/a = \nabla p$ with $p \in H^1(\Omega)$, then we have (cf. [6, Theorem 2.9])

$$\nabla \times (\mathbf{u}/a) \equiv \partial_1(u_2/a) - \partial_2(u_1/a) = 0\,, \text{ in } \Omega\,.$$

Moreover, the homogeneous Dirichlet boundary condition on Γ_D implies the tangential flux condition

$$\mathbf{n} \times (\mathbf{u}/a) \equiv (n_1u_2 - n_2u_1)/a = 0\,, \text{ on } \Gamma_D\,.$$

Adding these equations to first-order system (5) yields the augmented system

$$\begin{array}{rcll} \mathbf{u} - a\nabla p & = & \mathbf{0}\,, & \text{in } \Omega\,, \\ -\nabla\cdot\mathbf{u} & = & f\,, & \text{in } \Omega\,, \\ \nabla\times(\mathbf{u}/a) & = & 0\,, & \text{in } \Omega\,, \\ p & = & 0\,, & \text{on } \Gamma_D\,, \\ \mathbf{n}\cdot\mathbf{u} & = & 0\,, & \text{on } \Gamma_N\,, \\ \mathbf{n}\times(\mathbf{u}/a) & = & 0\,, & \text{on } \Gamma_D\,. \end{array} \tag{6}$$

In addition to $L^2(\Omega)$ and $H^1(\Omega)$ with the respective norms $\|\cdot\|_{0,\Omega}$ and $\|\cdot\|_{1,\Omega}$, we will need the spaces

$$H(\text{div};\Omega) = \{\mathbf{v} \in L^2(\Omega)^2 : \nabla\cdot\mathbf{v} \in L^2(\Omega)\}\,,$$
$$H(\text{curl}\,a;\Omega) = \{\mathbf{v} \in L^2(\Omega)^2 : \nabla\times(\mathbf{v}/a) \in L^2(\Omega)\}$$

and

$$\begin{array}{c} V = \{q \in H^1(\Omega) : q = 0 \text{ on } \Gamma_D\}\,, \\ \mathbf{W} = \{\mathbf{v} \in H(\text{div};\Omega) \cap H(\text{curl}\,a;\Omega) : \mathbf{n}\cdot\mathbf{v} = 0 \text{ on } \Gamma_N\,,\ \mathbf{n}\times(\mathbf{v}/a) = 0 \text{ on } \Gamma_D\}\,. \end{array} \tag{7}$$

Clearly, for the solution of (3), we have $p \in V$ and $\mathbf{u} \in \mathbf{W}$, so it is appropriate to pose (6) on these spaces.

As mentioned above, our main interest is in the solution of (3) when $a(x_1,x_2)$ has large jumps. Following Bramble, Pasciak, Wang, and Xu [2], we assume that

$$\overline{\Omega} = \bigcup_{i=1}^{J} \overline{\Omega}_i$$

with $\{\Omega_i\}$ being mutually disjoint open polygonal regions; that the restriction of $a(x_1,x_2)$ to Ω_i is in $C^1(\Omega_i)$; and that

$$c_1\omega_i \le a(x_1,x_2) \le c_2\omega_i \quad \text{for } (x_1,x_2) \in \Omega_i$$

with constants c_1, c_2 of order one and arbitrary positive constants ω_i. In other words, $a(x_1,x_2)$ is assumed to be of approximate size ω_i throughout Ω_i for each i while large variations in $\{\omega_i\}$ over i are allowed. The bounds derived below will be independent of this variation in $\{\omega_i\}$, but the constants in these bounds will depend on the variation within each Ω_i, that is, on c_1 and c_2.

An appropriate scaling of the equations in (6) leads to the least-squares functional

$$G(\mathbf{u},p;f) = \|\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p\|_{0,\Omega}^2 + \|\nabla\cdot\mathbf{u} + f\|_{0,\Omega}^2 + \|a\,\nabla\times(\mathbf{u}/a)\|_{0,\Omega}^2 \tag{8}$$

and associated bilinear form

$$\begin{array}{l} \mathcal{F}(\mathbf{u},p;\mathbf{v},q) = (\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p, \mathbf{v}/\sqrt{a} - \sqrt{a}\nabla q)_{0,\Omega} \\ +(\nabla\cdot\mathbf{u}, \nabla\cdot\mathbf{v})_{0,\Omega} + (a\,\nabla\times(\mathbf{u}/a), a\,\nabla\times(\mathbf{v}/a))_{0,\Omega}\,. \end{array} \tag{9}$$

Here, for the sake of notational simplicity, we agree that $(\cdot,\cdot)_{0,\Omega}$ is meant componentwise for vector functions. That is, if $\mathbf{w} = (w_1,w_2)$ and $\mathbf{z} = (z_1,z_2)$, then

$$(\mathbf{w},\mathbf{z})_{0,\Omega} = (w_1,z_1)_{0,\Omega} + (w_2,z_2)_{0,\Omega}\,.$$

The solution of (5) will also solve the minimization problem

$$G(\mathbf{u},p;f) = \min_{(\mathbf{v},q)\in \mathbf{W}\times V} G(\mathbf{v},q;f) \tag{10}$$

and, therefore, the variational problem

$$\mathcal{F}(\mathbf{u},p;\mathbf{v},q) = -(f,\nabla\cdot\mathbf{v})_{0,\Omega} \text{ for all } (\mathbf{v},q)\in\mathbf{W}\times V\,. \tag{11}$$

Here we show that $\mathcal{F}(\mathbf{v},q;\mathbf{v},q)$ is uniformly equivalent to the scaled norm defined for $(\mathbf{v},q)\in\mathbf{W}\times V$ by

$$|||(\mathbf{v},q)||| \equiv \left(||\nabla\cdot\mathbf{v}||^2_{0,\Omega} + ||a\nabla\times(\mathbf{v}/a)||^2_{0,\Omega} + ||\mathbf{v}/\sqrt{a}||^2_{0,\Omega} + ||\sqrt{a}\nabla q||^2_{0,\Omega}\right)^{1/2}\,.$$

Theorem 1 *Under the above assumptions, there exist constants γ_1 and γ_2, independent of the size of the jumps in $\{\omega_i\}$, such that*

$$\mathcal{F}(\mathbf{u},p;\mathbf{u},p)\geq\gamma_1|||(\mathbf{u},p)|||^2 \quad \textit{for all}\ (\mathbf{u},p)\in\mathbf{W}\times V \tag{12}$$

and

$$\mathcal{F}(\mathbf{u},p;\mathbf{v},q)\leq\gamma_2|||(\mathbf{u},p)|||\,|||(\mathbf{v},q)||| \quad \textit{for all}\ (\mathbf{u},p)\,,\ (\mathbf{v},q)\in\mathbf{W}\times V\,. \tag{13}$$

Proof. The proof is similar to the proof of [4, Theorem 3.1] (see also [10, Theorems 2.1 and 2.2]). We include it here because we must confirm that the constants γ_1 and γ_2 are independent of the jumps in a. The main part of the proof consists in showing that the functionals

$$\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{v},q) = (\mathbf{u}/\sqrt{a}-\sqrt{a}\nabla p, \mathbf{v}/\sqrt{a}-\sqrt{a}\nabla q)_{0,\Omega} + (\nabla\cdot\mathbf{u},\nabla\cdot\mathbf{v})_{0,\Omega}$$

and

$$\mathcal{S}(\mathbf{u},p;\mathbf{v},q) = (\mathbf{u}/\sqrt{a},\mathbf{v}/\sqrt{a})_{0,\Omega} + (\sqrt{a}\nabla p,\sqrt{a}\nabla q)_{0,\Omega} + (\nabla\cdot\mathbf{u},\nabla\cdot\mathbf{v})_{0,\Omega}\,,$$

satisfy

$$c_1\mathcal{S}(\mathbf{u},p;\mathbf{u},p)\leq\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p) \tag{14}$$

and

$$\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{v},q)\leq c_2(\mathcal{S}(\mathbf{u},p;\mathbf{u},p))^{1/2}(\mathcal{S}(\mathbf{v},q;\mathbf{v},q))^{1/2} \tag{15}$$

with constants c_1 and c_2 that are independent of the jumps in a.

For the proof of (14), we rewrite Poincaré-Friedrichs inequality (4) as

$$||p||^2_{0,\Omega}\leq\tilde{\gamma}||\sqrt{a}\nabla p||^2_{0,\Omega}\,. \tag{16}$$

Note that $\tilde{\gamma}$, and consequently the quantity γ_1 in (12), depends on $\min_{\mathbf{x}\in\Omega}a(\mathbf{x})>0$. It does not introduce, however, any dependence of (12) and (13) on the size of the jumps in a. Since on $\partial\Omega$ we either have $p=0$ or $\mathbf{n}\cdot\mathbf{u}=0$, then integration by parts confirms that

$$(\mathbf{u},\nabla p)_{0,\Omega} + (\nabla\cdot\mathbf{u},p)_{0,\Omega} = 0\,.$$

For any $\tau>0$, which we specify later, we have

$$\begin{aligned}
&\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p)\\
&= (\mathbf{u}/\sqrt{a},\mathbf{u}/\sqrt{a})_{0,\Omega} + (\sqrt{a}\nabla p,\sqrt{a}\nabla p)_{0,\Omega} - 2(\mathbf{u},\nabla p)_{0,\Omega} + (\nabla\cdot\mathbf{u},\nabla\cdot\mathbf{u})_{0,\Omega}\\
&\quad +2\tau(\nabla\cdot\mathbf{u},p)_{0,\Omega} + 2\tau(\mathbf{u},\nabla p)_{0,\Omega} + \tau^2(p,p)_{0,\Omega} - \tau^2(p,p)_{0,\Omega}\\
&= (\mathbf{u}/\sqrt{a}+(\tau-1)\sqrt{a}\nabla p,\mathbf{u}/\sqrt{a}+(\tau-1)\sqrt{a}\nabla p)_{0,\Omega}\\
&\quad +(\nabla\cdot\mathbf{u}+\tau p,\nabla\cdot\mathbf{u}+\tau p)_{0,\Omega} - \tau^2(p,p)_{0,\Omega} + (2\tau-\tau^2)(\sqrt{a}\nabla p,\sqrt{a}\nabla p)_{0,\Omega}\\
&\geq (2\tau-\tau^2)(\sqrt{a}\nabla p,\sqrt{a}\nabla p)_{0,\Omega} - \tau^2(p,p)_{0,\Omega}\\
&\geq (2\tau-(1+\tilde{\gamma})\tau^2)||\sqrt{a}\nabla p||^2_{0,\Omega}\,.
\end{aligned}$$

Choosing $\tau = 1/(1+\tilde{\gamma})$ leads to

$$\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p) \geq \tau \|\sqrt{a}\nabla p\|_{0,\Omega}^2 .$$

We then also have

$$\|\mathbf{u}/\sqrt{a}\|_{0,\Omega}^2 \leq 2(\|\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p\|_{0,\Omega}^2 + \|\sqrt{a}\nabla p\|_{0,\Omega}^2) \leq 2(1+1/\tau)\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p)$$

and, clearly,

$$\|\nabla \cdot \mathbf{u}\|_{0,\Omega}^2 \leq \hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p) ,$$

which completes the proof of (14).

Upper bound (15) follows from

$$\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{v},q) \leq 2(\hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p))^{1/2}(\hat{\mathcal{F}}(\mathbf{v},q;\mathbf{v},q))^{1/2}$$

and

$$\begin{aligned} \hat{\mathcal{F}}(\mathbf{u},p;\mathbf{u},p) &= \|\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p\|_{0,\Omega}^2 + \|\nabla \cdot \mathbf{u}\|_{0,\Omega}^2 \\ &\leq 2(\|\mathbf{u}/\sqrt{a}\|_{0,\Omega}^2 + \|\sqrt{a}\nabla p\|_{0,\Omega}^2 + \|\nabla \cdot \mathbf{u}\|_{0,\Omega}^2) = \mathcal{S}(\mathbf{u},p;\mathbf{u},p) . \end{aligned} \qquad (17).$$

The proof of Theorem 1 is completed by adding the term $\|a\nabla \times (\mathbf{u}/a)\|_{0,\Omega}$ to both sides of the inequalities (14) and (17). ▮

Theorem 1 states that ellipticity and continuity of the least-squares bilinear form $\mathcal{F}(\cdot,\cdot;\cdot,\cdot)$ in terms of the norm $|||(\cdot,\cdot)|||$ is independent of the jumps in a. Note, however, that the ellipticity constant γ_1 in (12) depends on the size of a, in particular, on the positive constant $\min_{\mathbf{x}\in\Omega} a(\mathbf{x})$ through the Poincaré-Friedrichs inequality (16).

The scaling of the norm $|||(\cdot,\cdot)|||$ has the following physical interpretation. In areas where a is relatively small, ∇p is allowed to be relatively large, and one has to expect a less accurate approximation there compared to areas where a is large and ∇p is therefore small. In contrast, the velocity $\mathbf{u} = a\nabla p$ can be expected to be more accurate in areas where a is small and less accurate, in general, where a is large. Ellipticity with constants that are independent of the jumps in a asserts that the scaling in $\mathcal{F}(\cdot,\cdot;\cdot,\cdot)$ correctly reflects these attributes.

Singularities at Interface Corners and Cross-Points

This section is concerned with the behavior of p and $\mathbf{u}$ at or near the interface curve. Most of what we present in this section is well-known; we refer to Strang and Fix [14, Chapter 8] for further details.

Recall from the previous section that the solution of (6) satisfies $\mathbf{u} \in H(\text{div};\Omega) \cap H(\text{curl } a;\Omega)$. This implies that, at a point on a smooth segment of the interface curve, the normal component $\mathbf{n}\cdot\mathbf{u}$ and the tangential component $\mathbf{n} \times (\mathbf{u}/a)$ must be continuous. Assume that $\overline{\Omega} = \overline{\Omega}^+ \cup \overline{\Omega}^-$ with constant diffusion coefficients a^+ and a^-, respectively, and let $\mathbf{u}^+ = (u_1^+, u_2^+)$ and $\mathbf{u}^- = (u_1^-, u_2^-)$ denote the solution restricted to the respective subdomains (see Figure 1). Then u_1 and u_2 must satisfy the jump conditions

$$n_1 u_1^+ + n_2 u_2^+ = n_1 u_1^- + n_2 u_2^- \text{ and } n_2 \frac{u_1^+}{a^+} - n_1 \frac{u_2^+}{a^+} = n_2 \frac{u_1^-}{a^-} - n_1 \frac{u_2^-}{a^-} . \qquad (18)$$

For example, consider the situation shown in Figure 1 (which we will encounter again as Example 2 in the final section of this paper). Across the vertical part of the interface, $u_1 = \mathbf{n}\cdot\mathbf{u}$ will be continuous while $u_2 = \mathbf{n} \times \mathbf{u}$ has a jump factor of a^+/a^-. Similarly, across the horizontal part of the interface, $u_1 = -\mathbf{n} \times \mathbf{u}$ has a jump factor of a^+/a^- while $u_2 = \mathbf{n}\cdot\mathbf{u}$ is continuous. At the interface corner, both of these conditions must be satisfied, i.e., u_1 and u_2 must jump by a factor a^+/a^- and be continuous at the same

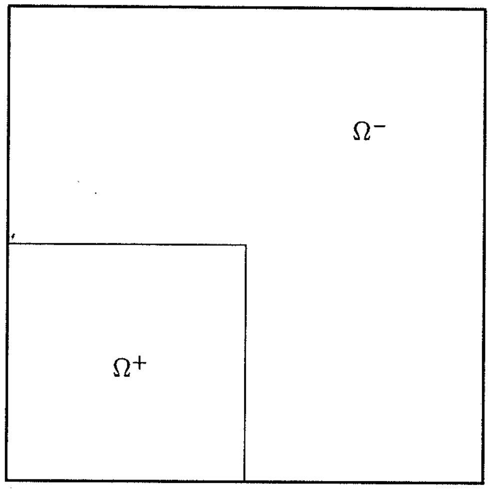


Figure 1: Interface with corner

time. Obviously, there are only two ways for this to happen: either $\mathbf{u} = \mathbf{0}$ or $\mathbf{u} = \infty$ at the interface corner. In general, the latter case is encountered at interface corners—the behavior of $\mathbf{u}$ is singular there.

Without loss of generality, assume that the singularity occurs at the origin, and consider the polar coordinate representation

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} r\ \cos\theta \\ r\ \sin\theta \end{pmatrix}$$

The solution of (3) then admits the representation

$$p(r,\theta) = \begin{cases} r^\alpha(\lambda_c^+ \cos\alpha\theta + \lambda_s^+ \sin\alpha\theta) + \tilde{p}^+(r,\theta)\ , & \text{in } \Omega^+\ , \\ r^\alpha(\lambda_c^- \cos\alpha\theta + \lambda_s^- \sin\alpha\theta) + \tilde{p}^-(r,\theta)\ , & \text{in } \Omega^-\ , \end{cases}$$

where $\tilde{p}^+ \in H^2(\Omega^+), \tilde{p}^- \in H^2(\Omega^-)$ (cf. [14, Section 8.1]), $\alpha \in (1/2, 1)$, and $\lambda_c^\pm, \lambda_s^\pm$ are constants. Using

$$\nabla = \begin{pmatrix} \partial_1 \\ \partial_2 \end{pmatrix} = \begin{pmatrix} \cos\theta\ \frac{\partial}{\partial r} - \sin\theta\ \frac{1}{r}\ \frac{\partial}{\partial\theta} \\ \sin\theta\ \frac{\partial}{\partial r} + \cos\theta\ \frac{1}{r}\ \frac{\partial}{\partial\theta} \end{pmatrix} \tag{19}$$

leads to

$$u_1(r,\theta) = \begin{cases} \alpha a^+ r^{\alpha-1}(\lambda_c^+ \cos(\alpha-1)\theta + \lambda_s^+ \sin(\alpha-1)\theta) + \tilde{u}_1^+(r,\theta)\ , & \text{in } \Omega^+\ , \\ \alpha a^- r^{\alpha-1}(\lambda_c^- \cos(\alpha-1)\theta + \lambda_s^- \sin(\alpha-1)\theta) + \tilde{u}_1^-(r,\theta)\ , & \text{in } \Omega^-\ , \end{cases} \tag{20}$$

and

$$u_2(r,\theta) = \begin{cases} \alpha a^+ r^{\alpha-1}(-\lambda_c^+ \sin(\alpha-1)\theta + \lambda_s^+ \cos(\alpha-1)\theta) + \tilde{u}_2^+(r,\theta)\ , & \text{in } \Omega^+\ , \\ \alpha a^- r^{\alpha-1}(-\lambda_c^- \sin(\alpha-1)\theta + \lambda_s^- \cos(\alpha-1)\theta) + \tilde{u}_2^-(r,\theta)\ , & \text{in } \Omega^-\ , \end{cases} \tag{21}$$

with $\tilde{u}_1^+, \tilde{u}_2^+ \in H^1(\Omega^+)$ and $\tilde{u}_1^-, \tilde{u}_2^- \in H^1(\Omega^-)$. The parameters $\alpha, \lambda_c^+, \lambda_s^+, \lambda_c^-$, and λ_s^- are computed such that conditions (18) are fulfilled. Setting $\mu = a^+/a^-$ leads to the matrix equation

$$\begin{bmatrix} -\mu\sin\alpha\frac{3}{2}\pi & \mu\cos\alpha\frac{3}{2}\pi & -\sin\alpha\frac{\pi}{2} & -\cos\alpha\frac{\pi}{2} \\ -\cos\alpha\frac{3}{2}\pi & -\sin\alpha\frac{3}{2}\pi & \cos\alpha\frac{\pi}{2} & -\sin\alpha\frac{\pi}{2} \\ -\cos\alpha\pi & -\sin\alpha\pi & \cos\alpha\pi & \sin\alpha\pi \\ \mu\sin\alpha\pi & -\mu\cos\alpha\pi & -\sin\alpha\pi & \cos\alpha\pi \end{bmatrix} \begin{bmatrix} \lambda_c^+ \\ \lambda_s^+ \\ \lambda_c^- \\ \lambda_s^- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

For this homogeneous system of linear equations to have a nontrivial solution, its determinant must vanish, which leads to

$$\frac{1}{2}(\mu+\frac{1}{\mu})(\cos\pi\alpha-\cos 2\pi\alpha)+2-\cos\pi\alpha-\cos 2\pi\alpha=0\,. \tag{22}$$

The exponent α that determines the degree of the singularity apparently depends on the size of the jump μ. It can be shown that (22) always has a unique solution $\alpha \in (1/2, 1)$. For $\mu \longrightarrow 1$, i.e., as the jump disappears, we have $\alpha \longrightarrow 1$, i.e., the singularity disappears as well. For $\mu \longrightarrow 0$ or $\mu \longrightarrow \infty$, α tends to $2/3$, which is exactly the value obtained for a reentrant corner with exterior angle $\pi/2$. It is straightforward to extend the procedure outlined above to any number of adjoining subdomains and any size of angles (cf. [8]). We therefore have a computational technique to compute the shape of the singularity at interface corners and cross-points where two interfaces intersect. This technique will be fundamental for the finite element approach described in the next section.

Finite Element Approximation

The minimum of $G(\mathbf{u}, p; f)$ is approximated using a Rayleigh-Ritz finite element method. Let $\mathcal{T}^h$ be a triangulation of Ω, which we assume to be quasi-uniform (cf. [3, Chapter 4]), and let $\mathbf{W}^h$ and V^h be appropriate finite-dimensional spaces. The interface is required to be the union of edges of the triangulation. If the interface is cutting through elements of the triangulation, then special techniques have to be considered in order to average the parameters properly, which complicates the whole approach. We do not address this task or the problems associated with it here, but instead assume that the interfaces are restricted to edges of the triangulation. For the sake of exposition, we also assume that each segment of the interface curves is parallel to one of the coordinate axes. It is easy to see that the following development of the finite element approach can be generalized to isoparametric elements, where the interface curves are logically aligned with coordinate axes.

It is desirable, in general, to use conforming finite elements, where the finite-dimensional spaces satisfy $\mathbf{W}^h \subset \mathbf{W}$ and $V^h \subset V$. Along straight segments of the interface curve, this can be accomplished by enforcing condition (18) on the finite element basis functions. Using bilinear finite elements on rectangles, for example, a basis function for u_1 at a node on a horizontal interface segment is continuous in the x_1-direction and has a jump of size a^+/a^- in the x_2-direction. Such a basis function for u_1 at a node on a vertical interface segment is continuous (in both coordinate directions). Under the assumption that all the interface curves are straight lines which do not intersect each other (we will address the case of interface corners or cross-points later), we can therefore construct piecewise bilinear finite element spaces:

$$\begin{aligned}&V^h=\{q\in V : q|_T \text{ bilinear on } T \text{ for all } T\in\mathcal{T}^h\}\\&\mathbf{W}^h=\{\mathbf{v}\in H(\operatorname{div},\Omega)\cap H(\operatorname{curl} a,\Omega) : v_i|_T \text{ bilinear on } T \text{ for all } T\in\mathcal{T}^h\}\,.\end{aligned}$$

The finite element approximation $(\mathbf{u}^h, p^h) \in \mathbf{W}^h \times V^h$ is then defined as the solution of the minimization problem

$$G(\mathbf{u}^h,p^h;f)=\min_{(\mathbf{v}^h,q^h)\in\mathbf{W}^h\times V^h} G(\mathbf{v}^h,q^h;f)\,. \tag{23}$$

One of the main practical advantages of the least-squares finite element approach over other variational formulations consists in the fact that the minimum of the functional constitutes an a posteriori error measure. This follows from the general relation between the least-squares functional and corresponding bilinear form. The main point here is the fact that the least-squares functional is zero at the solution $(\mathbf{u}, p)$, which leads to

$$\begin{aligned} & G(\mathbf{u}^h, p^h; f) \\ = \; & G(\mathbf{u}^h, p^h; f) - G(\mathbf{u}, p; f) \\ = \; & \mathcal{F}(\mathbf{u}^h, p^h; \mathbf{u}^h, p^h) + 2(f, \nabla \cdot \mathbf{u}^h)_{0,\Omega} - \mathcal{F}^h(\mathbf{u}, p; \mathbf{u}, p) - 2(f, \nabla \cdot \mathbf{u})_{0,\Omega} \\ = \; & \mathcal{F}^h(\mathbf{u} - \mathbf{u}^h, p - p^h; \mathbf{u} - \mathbf{u}^h, p - p^h) \,. \end{aligned}$$

Under the above assumptions, we get the following convergence result for the finite element approximation.

Theorem 2 *Assume that for* $(\mathbf{u}, p)$, *the solution of* (10), *we have* $(\mathbf{u}, p)|_{\Omega_i} \in (H^{1+\delta}(\Omega_i))^3$ *for some* $\delta \in (0, 1]$ *and for* $i = 1, \ldots, J$. *Let* $(\mathbf{u}^h, p^h) \in \mathbf{W}^h \times V^h$ *be the solution of* (23). *Then*

$$|||(\mathbf{u}, p) - (\mathbf{u}^h, p^h)||| \le Ch^\delta \sum_{i=1}^{J} (||\mathbf{u}||_{1+\delta,\Omega_i} + ||\sqrt{\omega_i} p||_{1+\delta,\Omega_i}) \tag{24}$$

where the constant C *is independent of* h *and of the size of the jumps in* $\{\omega_i\}$.

Proof. From Theorem 1 and Cea's Lemma (see, for example, [3, Theorem 2.8.1]), we obtain

$$|||(\mathbf{u}, p) - (\mathbf{u}^h, p^h)||| \le \frac{\gamma_2}{\gamma_1} \min_{(\mathbf{v}^h, q^h) \in \mathbf{W}^h \times V^h} |||(\mathbf{u}, p) - (\mathbf{v}^h, q^h)||| \,.$$

Moreover, for $(\mathbf{v}, q) \in \mathbf{W} \times V$, we have

$$\begin{aligned} |||(\mathbf{v}, q)|||^2 &= ||\nabla \cdot \mathbf{v}||^2_{0,\Omega} + ||a \nabla \times (\mathbf{v}/a)||^2_{0,\Omega} + ||\mathbf{v}/\sqrt{a}||^2_{0,\Omega} + ||\sqrt{a} \nabla q||^2_{0,\Omega} \\ &= \sum_{i=1}^{J} (||\nabla \cdot \mathbf{v}||^2_{0,\Omega_i} + ||a \nabla \times (\mathbf{v}/a)||^2_{0,\Omega_i} + ||\mathbf{v}/\sqrt{a}||^2_{0,\Omega_i} + ||\sqrt{a} \nabla q||^2_{0,\Omega_i}) \\ &\le c_1 \sum_{i=1}^{J} (||\nabla \cdot \mathbf{v}||^2_{0,\Omega_i} + ||\nabla \times \mathbf{v}||^2_{0,\Omega_i} + ||\mathbf{v}/\sqrt{a}||^2_{0,\Omega_i} + ||\sqrt{a} \nabla q||^2_{0,\Omega_i}) \,. \end{aligned}$$

Since by assumption $\mathbf{u}|_{\Omega_i} \in H^1(\Omega_i)$ and, similarly, $\mathbf{v}^h|_{\Omega_i} \in H^1(\Omega_i)$ for each $\mathbf{v}^h \in \mathbf{W}^h$, then for $i = 1, \ldots, J$ we have

$$||\nabla \cdot (\mathbf{u} - \mathbf{v}^h)||^2_{0,\Omega_i} + ||\nabla \times (\mathbf{u} - \mathbf{v}^h)||^2_{0,\Omega_i} \le c_2 |\mathbf{u} - \mathbf{v}^h|^2_{1,\Omega_i} \,.$$

This leads to

$$|||(\mathbf{u}, p) - (\mathbf{v}^h, q^h)||| \le c_3 \sum_{i=1}^{J} (|\mathbf{u} - \mathbf{v}^h|_{1,\Omega_i} + ||(\mathbf{u} - \mathbf{v}^h)/\sqrt{\omega_i}||_{0,\Omega_i} + ||\sqrt{\omega_i}(p - q^h)||_{1,\Omega_i}) \,.$$

Standard interpolation properties of piecewise bilinear functions (see, for example, [3, Theorems 12.3.3 and 12.3.12]) lead to

$$\begin{aligned} ||\mathbf{u} - \mathbf{v}^h||_{1,\Omega_i} &\le c_4 h^\delta ||\mathbf{u}||_{1+\delta,\Omega_i} \\ ||p - q^h||_{1,\Omega_i} &\le c_5 h^\delta ||p||_{1+\delta,\Omega_i} \end{aligned}$$

which completes the proof. ▮

If the interface curve is not a straight line, or, more generally, not sufficiently smooth, then the finite element approximation becomes excessively more complicated. In the preceding section we saw that, for the solution $(\mathbf{u}, p)$ of (10), $\mathbf{u}$ has the singular behavior shown in (20) and (21). It is easy to see that this implies $\mathbf{u}|_{\Omega_i} \notin (H^1(\Omega_i))^2$ for all subregions Ω_i adjacent to the interface corner, and therefore the standard finite element approximation results do not apply.

Moreover, in order to have $\mathbf{u}^h \in H(\mathrm{div}, \Omega) \cap H(\mathrm{curl}\, a, \Omega)$ in the neighborhood of an interface corner, it is necessary and sufficient to require $\mathbf{u}^h$ to have the form of (20) and (21). In other words, in order to have conforming finite elements, we must include a singular basis function at each interface corner (or cross-point). The tools developed in the previous section allow us, in principle, to compute the exact shape of such a singularity. Multiplied by a standard piecewise bilinear function, such a singular function could then serve as a basis function at that point. A procedure of this type is described in [14, Section 8.2] along with special techniques to solve the resulting discrete system. However, this approach requires special stencils for these singular points, which complicates the overall finite element approach. Instead, we consider an alternative nonconforming finite element method, based on simple basis functions like bilinears on rectangles.

We construct $\mathbf{W}^h$ observing the fact that, for the right-hand side in (11) to be defined, we must have $\mathbf{W}^h \subset H(\mathrm{div}, \Omega)$. This implies that, for $\mathbf{u}^h \in \mathbf{W}^h$, $\mathbf{n} \cdot \mathbf{u}^h$ must be continuous across all interfaces. Now consider the bilinear finite element basis function associated with the interface corner in Figure 1. For $\mathbf{u}^h \in \mathbf{W}^h \subset H(\mathrm{div}, \Omega)$, we must require that u_1 is continuous in the x_1-direction across the horizontal portion of the interface; that u_2 is continuous in the x_2-direction across the vertical portion of the interface; and that both u_1, u_2 are continuous elsewhere. From (18) we see that $\mathbf{u} \in H(\mathrm{curl}\, a, \Omega)$ requires u_2 to have a jump across the vertical portion of the interface, while u_1 must have a jump across the horizontal portion. This causes a conflict at the corner. The finite-dimensional space $\mathbf{W}^h$ will, therefore, not be contained in $H(\mathrm{curl}\, a, \Omega)$, in general, and $\mathbf{W}^h \times V^h \not\subset \mathbf{W} \times V$. In particular, the bilinear form $\mathcal{F}(\cdot, \cdot; \cdot, \cdot)$ is not defined on $\mathbf{W}^h \times V^h$. For $\mathbf{u}, \mathbf{v} \in \mathbf{W} + \mathbf{W}^h$ and $p, q \in V + V^h$, we define a modified least-squares bilinear form by

$$\begin{aligned} \mathcal{F}^h(\mathbf{u}, p; \mathbf{v}, q) &= (\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p, \mathbf{v}/\sqrt{a} - \sqrt{a}\nabla q)_{0,\Omega} \\ &+ (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v})_{0,\Omega} + \textstyle\sum_{i=1}^{J} (\nabla \times \mathbf{u}, \nabla \times \mathbf{v})_{0,\Omega_i} \,. \end{aligned} \tag{25}$$

On $\mathbf{W} \times V$, this bilinear form coincides with $\mathcal{F}(\cdot, \cdot; \cdot, \cdot)$. The least-squares functional corresponding to $\mathcal{F}^h(\cdot, \cdot; \cdot, \cdot)$ is

$$G^h(\mathbf{u}, p; f) = \|\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p\|_{0,\Omega}^2 + \|\nabla \cdot \mathbf{u} + f\|_{0,\Omega}^2 + \sum_{i=1}^{J} \|\nabla \times \mathbf{u}\|_{0,\Omega_i}^2 \,. \tag{26}$$

Let $(\mathbf{u}, p) \in \mathbf{W} \times V$ be the solution of (10), and let $(\mathbf{u}^h, p^h) \in \mathbf{W}^h \times V^h$ be defined by

$$G^h(\mathbf{u}^h, p^h; f) = \min_{(\mathbf{v}^h, q^h) \in \mathbf{W}^h \times V^h} G^h(\mathbf{v}^h, q^h; f) \,. \tag{27}$$

Recall that, at an interface corner, $\mathbf{u}$ has a singularity of the form given in (20) and (21). This implies that we cannot expect to approximate $\mathbf{u}$ to the same accuracy by standard finite elements near a singularity as elsewhere in Ω. Moreover, since our finite element subspace $\mathbf{W}^h \times V^h$ is not contained in the space $\mathbf{W} \times V$ in which we have shown ellipticity, the relatively large error near a singularity will deterioriate the finite element approximation in the entire region. This phenomenon is reflected by the fact that, in the presence of singularities, $G^h(\mathbf{u}^h, p^h; f)$ does not decrease as h is made smaller. We will observe this behavior later in our computational experiments. It is therefore necessary to introduce a weight function which decreases near the singular point. The proper choice of weighting is motivated by the form of the singularity.

In particular, (19), (20), and (21) imply $\nabla \mathbf{u} \sim r^{\alpha - 2}$ in the neighborhood of the singularity. If T_s^h denotes an element of the triangulation $\mathcal{T}^h$ such that the interface corner appears as one of its vertices, then

$$\|r^{2-\alpha}\nabla(u_j - u_j^h)\|_{0,T_s^h} = O(h^2) \,.$$

If the right-hand side f and the restriction of a to Ω_i are sufficiently smooth, then we know that $\mathbf{u} \in (H^2_{\mathrm{loc}}(\Omega_i))^2$, i.e., $\mathbf{u} \in (H^2(\tilde{\Omega}))^2$ for any compact $\tilde{\Omega} \subset \Omega_i$. This implies that $\tilde{\mathbf{v}}^h \in \mathbf{W}^h$ exists such that

$$\|\nabla \cdot (\mathbf{u} - \tilde{\mathbf{v}}^h)\|_{0,\tilde{\Omega}} = O(h^2)\,.$$

The other terms in (26) can be treated in a similar way, which motivates the definition of the weighted least-squares functional

$$\begin{array}{l} G^h_w(\mathbf{u}, p; f) = \|\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p\|^2_{0,h,1-\alpha,\Omega} \\ + \|\nabla \cdot \mathbf{u} + f\|^2_{0,h,2-\alpha,\Omega} + \sum_{i=1}^J \|\nabla \times \mathbf{u}\|^2_{0,h,2-\alpha,\Omega_i} \end{array} \tag{28}$$

and corresponding bilinear form

$$\begin{array}{l} \mathcal{F}^h_w(\mathbf{u}, p; \mathbf{v}, q) = (\mathbf{u}/\sqrt{a} - \sqrt{a}\nabla p), \mathbf{v}/\sqrt{a} - \sqrt{a}\nabla q))_{0,h,1-\alpha,\Omega} \\ + (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v})_{0,h,2-\alpha,\Omega} + \sum_{i=1}^J (\nabla \times \mathbf{u}). \nabla \times \mathbf{v})_{0,h,2-\alpha,\Omega_i}\,. \end{array} \tag{29}$$

The inner product $(\cdot,\cdot)_{0,h,\beta,\Omega}$ is defined as

$$(\mathbf{v}, \mathbf{w})_{0,h,\beta,\Omega} = (w^{h,\beta}\mathbf{v}, w^{h,\beta}\mathbf{w})_{0,\Omega}$$

with the weight function $w^{h,\beta}$ constructed in the following way: Consider a sequence of triangulations $\{\mathcal{T}^{h_l}, l = 0, \ldots, L\}$, with $H = h_0 \geq h_1 \geq \cdots \geq h_L = h$. Let $\Omega^{h_l}_s$ denote the union of of all elements $T^{h_l} \in \mathcal{T}^{h_l}$ with the singular point as one of their vertices. The weight function $w^{h,\beta}$ is defined as

$$w^{h,\beta}(\mathbf{x}) = \begin{cases} h^\beta \text{ for } \mathbf{x} \in \Omega^h_s\,, \\ h^\beta_l \text{ for } \mathbf{x} \in \Omega^{h_{l-1}}_s \backslash \Omega^{h_l}_s\,, \; l = 1, \ldots, L\,, \\ 1 \text{ for } \mathbf{x} \in \Omega \backslash \Omega^{h_0}_s\,. \end{cases} \tag{30}$$

Let $(\mathbf{u}^h_w, p^h_w) \in \mathbf{W}^h \times V^h$ be the solution of

$$G^h_w(\mathbf{u}^h_w, p^h_w; f) = \min_{(\mathbf{v}^h, q^h) \in \mathbf{W}^h \times V^h} G^h_w(\mathbf{v}^h, q^h; f)\,. \tag{31}$$

In the final section of this paper we will demonstrate, by means of numerical results, that the weighted functional $G^h_w(\mathbf{u}^h_w, p^h_w; f)$ actually decreases regularly as the triangulation is refined. Note, however, that this does not mean that the error $\mathbf{u} - \mathbf{u}^h_w$ is small throughout the region Ω. In particular, the pointwise accuracy usually deteriorates near singularities. This suggests that the weighted functional should be combined with local refinement techniques to guarantee satisfactory resolution in the entire region. Multilevel refinement techniques are especially effective in this context.

Multilevel Algorithms

Consider the sequence of triangulations $\{\mathcal{T}^{h_l}, l = 0, \ldots, L\}$ introduced earlier. Associated with each triangulation $\{\mathcal{T}^{h_l}\}$ is the finite element space $\mathbf{W}^{h_l} \times V^{h_l}$, which we may also denote by $\mathbf{W}_l \times V_l$. This leads to a nested sequence of spaces

$$\mathbf{W}_0 \times V_0 \subset \mathbf{W}_1 \times V_1 \subset \cdots \subset \mathbf{W}_L \times V_L = \mathbf{W}^h \times V^h\,.$$

On each level l, $0 \leq l \leq L$, an operator $\mathcal{F}_l : \mathbf{W}_l \times V_l \to \mathbf{W}_l \times V_l$ is defined by

$$\langle\langle \mathcal{F}_l(\mathbf{u}, p); (\mathbf{v}, q) \rangle\rangle = \mathcal{F}(\mathbf{u}, p; \mathbf{v}, q) \text{ for all } (\mathbf{v}, q) \in \mathbf{W}_l \times V_l\,,$$

where the inner product $\langle\langle \cdot; \cdot \rangle\rangle$ is given by

$$\langle\langle (\mathbf{u}, p); (\mathbf{v}, q) \rangle\rangle = (\mathbf{u}, \mathbf{v})_{0,\Omega} + (\sqrt{a}\, p, \sqrt{a}\, q)_{0,\Omega}\,.$$

In terms of the operator $\mathcal{F}_l$, the discrete problem (23) can be written as

$$\mathcal{F}_l(\mathbf{u}_l, p_l) = F_l \tag{32}$$

where the right-hand side is defined by $\langle\langle F_l, (\mathbf{v}, q)\rangle\rangle = -(f, \nabla\cdot\mathbf{v})_{0,\Omega}$ for all $(\mathbf{v}, q) \in \mathbf{W}_l \times V_l$. For the solution of (32), it is natural to use an iterative method since this requires only a computational procedure for the action of the operator $\mathcal{F}_l$ for $l = 0, \ldots, L$. The cost for one call of such a procedure is proportional to the number of unknowns $N = O(h^{-2})$.

The conjugate gradient method (cf. [13, Section 8.7]) computes its iterates $(\mathbf{u}_l^{(n)}, p_l^{(n)}) \in \mathbf{W}_l \times V_l$ in the Krylov subspace

$$\mathcal{K}_n(F_l, \mathcal{F}_l) = \operatorname{span}\{F_l, \mathcal{F}_l F_l, \ldots, \mathcal{F}_l^{n-1} F_l\}$$

according to the minimization property

$$G(\mathbf{u}_l^{(n)}, p_l^{(n)}; f) = \min_{(\mathbf{v}_l, q_l) \in \mathcal{K}_n(F_l, \mathcal{F}_l)} G(\mathbf{v}_l, q_l; f)\,.$$

Since the condition number of $\mathcal{F}_l$ is proportional to $O(h_l^{-2})$ (cf. [5, Theorem 3.2]), the number of conjugate gradient iterations required to achieve a certain accuracy grows like $O(h_l^{-1})$ (cf. [13, Section 8.7]). The overall computational complexity to solve a discrete problem on $\mathcal{T}^{h_l}$ using the conjugate gradient method therefore grows like $O(h_l^{-3})$.

Optimal computational complexity, $O(h_l^{-2})$, can be achieved, under certain assumptions on $\mathcal{F}((\cdot,\cdot);(\cdot,\cdot))$, by a full multigrid algorithm. The basic ingredients for multilevel methods are the projection operators $\mathcal{P}_l, \mathcal{Q}_l : \mathbf{W}^h \times V^h \to \mathbf{W}_l \times V_l$ which are given by

$$\mathcal{F}(\mathcal{P}_l(\mathbf{u}, p); (\mathbf{v}, q)) = \mathcal{F}((\mathbf{u}, p); (\mathbf{v}, q)) \text{ for all } (\mathbf{v}, q) \in \mathbf{W}_l \times V_l$$

and

$$\langle\langle \mathcal{Q}_l(\mathbf{u}, p); (\mathbf{v}, q)\rangle\rangle = \langle\langle (\mathbf{u}, p); (\mathbf{v}, q)\rangle\rangle \text{ for all } (\mathbf{v}, q) \in \mathbf{W}_l \times V_l$$

and smoothing operators $\mathcal{R}_l : \mathbf{W}_l \times V_l \to \mathbf{W}_l \times V_l$ representing iterations on level l. With these tools, standard multilevel algorithms can be constructed (see [5, Section 4] for further details). A detailed study of the convergence properties of multilevel methods for first-order system least-squares applied to problems with discontinuous coefficients will be given in [9].

Computational Experiments

In our examples, we consider (3) on the unit square $\Omega = \{(x_1, x_2) \in \mathbb{R}^2 : 0 < x_1, x_2 < 1\}$, with $f \equiv 1$ and $\Gamma_D = \partial\Omega$. We show the results of two sets of experiments, one with a smooth interface curve and the other with an interface corner causing a singularity in $\mathbf{u}$.

Example 1. In this example, the interface curve is a straight line, so no singularity occurs. We consider

$$a(x_1, x_2) = \begin{cases} a^+\,, & 0 < x_2 < 0.5\,, \\ a^-\,, & 0.5 < x_2 < 1\,, \end{cases} \tag{33}$$

with different choices for the values for a^+ and a^-. The solution shown in Figure 3 was obtained for $a^+ = 10$ and $a^- = 0.1$.

The computational results shown in Table 1 indicate that the approximation of the solution improves nicely as the triangulation is refined, independently of the size of the jumps. The reduction factor displayed in parentheses is the ratio of the minimum values on the current and next coarser level. Note that they do not quite reach 0.25, which is due to the lack of regularity at the corners of the subdomains. In fact, due to the corners

Table 1: Example 1: Minimum value (reduction factor) of the functional G^h

a^+/a^-	1	10	10^2	10^4
h				
1/8	$2.42 \cdot 10^{-2}$	$3.50 \cdot 10^{-2}$	$4.13 \cdot 10^{-2}$	$7.81 \cdot 10^{-2}$
1/16	$7.18 \cdot 10^{-3}$ (0.30)	$1.07 \cdot 10^{-2}$ (0.31)	$1.26 \cdot 10^{-2}$ (0.31)	$2.30 \cdot 10^{-2}$ (0.29)
1/32	$2.08 \cdot 10^{-3}$ (0.29)	$3.14 \cdot 10^{-3}$ (0.29)	$3.71 \cdot 10^{-3}$ (0.29)	$6.41 \cdot 10^{-3}$ (0.28)
1/64	$5.92 \cdot 10^{-4}$ (0.28)	$9.05 \cdot 10^{-4}$ (0.29)	$1.07 \cdot 10^{-3}$ (0.29)	$1.75 \cdot 10^{-3}$ (0.27)

with interior angle $\pi/2$, we have neither $\mathbf{u} \in (H^2(\Omega^+))^2$ nor $\mathbf{u} \in (H^2(\Omega^-))^2$. Consequently, the finite element approximation deteriorates near these corners. In contrast to the situation at singularities, however, this behavior does not contaminate the solution elsewhere since the basis functions corresponding to these points are conforming.

Example 2. This example shows results for a problem with a singularity in $\mathbf{u}$. We choose

$$a(x_1, x_2) = \begin{cases} a^+, & 0 < x_1, x_2 < 0.5\,, \\ a^-, & \text{elsewhere} \end{cases} \tag{34}$$

(see Figure 1) with different choices for the values for a^+ and a^- (again with $a^+ = 10$ and $a^- = 0.1$ for the solution shown in Figure 4).

The exponents for this example with the three values for the coefficient jumps used in Table 2 are given by $\alpha = 0.7317, 0.6739$, and 0.6667, respectively. Note that the last number is very close to the value $\alpha = 2/3$ that one gets for a reentrant corner with interior angle $3/2\pi$. Using the weighting described earlier with $H = 1/8$ leads to the results listed in Table 2. The modified least-squares functional is again reduced nicely and regularly as the triangulation is refined. Note that using the weighted functional means that the pointwise approximation deteriorates close to the singular point, where local refinement can be used if a better pointwise resolution is needed.

Table 2: Example 2: Minimum value (reduction factor) of the weighted functional G^h_w

a^+/a^-	1	10	10^2	10^4
h				
1/8	$2.42 \cdot 10^{-2}$	$3.74 \cdot 10^{-2}$	$5.17 \cdot 10^{-2}$	$1.20 \cdot 10^{-1}$
1/16	$7.18 \cdot 10^{-3}$ (0.30)	$1.16 \cdot 10^{-2}$ (0.31)	$1.58 \cdot 10^{-2}$ (0.31)	$3.53 \cdot 10^{-2}$ (0.29)
1/32	$2.08 \cdot 10^{-3}$ (0.29)	$3.43 \cdot 10^{-3}$ (0.30)	$4.66 \cdot 10^{-3}$ (0.29)	$9.84 \cdot 10^{-3}$ (0.28)
1/64	$5.92 \cdot 10^{-4}$ (0.28)	$9.95 \cdot 10^{-4}$ (0.29)	$1.34 \cdot 10^{-3}$ (0.29)	$2.68 \cdot 10^{-3}$ (0.27)

Table 3: Example 2: Minimum value of the functional G^h

a^+/a^-	1	10	10^2	10^4
h				
1/8	$2.42 \cdot 10^{-2}$	$4.36 \cdot 10^{-2}$	$7.50 \cdot 10^{-2}$	$1.62 \cdot 10^{-1}$
1/16	$7.18 \cdot 10^{-3}$	$2.39 \cdot 10^{-2}$	$5.49 \cdot 10^{-2}$	$9.89 \cdot 10^{-2}$
1/32	$2.08 \cdot 10^{-3}$	$2.07 \cdot 10^{-2}$	$5.35 \cdot 10^{-2}$	$8.86 \cdot 10^{-2}$
1/64	$5.92 \cdot 10^{-4}$	$2.22 \cdot 10^{-2}$	$5.66 \cdot 10^{-2}$	$9.33 \cdot 10^{-2}$

In order to illustrate the necessity of modifying the functional in the neighborhood of a singular point, we also computed the results for the unmodified functional G^h instead of G^h_w. The numbers in Table 3 show that this functional is not satisfactorily reduced in the course of refining the triangulation. Our numerical tests have shown that minimizing the unmodified functional leads to poor finite element approximations. Figure 2 shows the error with respect to the exact solution for p for the weighted functional and for the unmodified functional. Obviously, for the unmodified functional, the resulting error between the discrete and exact solution is relatively large in the entire domain. This behavior seems to indicate that using the unmodified functional has the effect of trying too hard to satisfy the first-order system (6) close to the singularity, where it is impossible to get a good approximation with bilinear finite elements. For the weighted functional, however, the error is smaller and mainly occurs in a rather small neighborhood of the singular point.

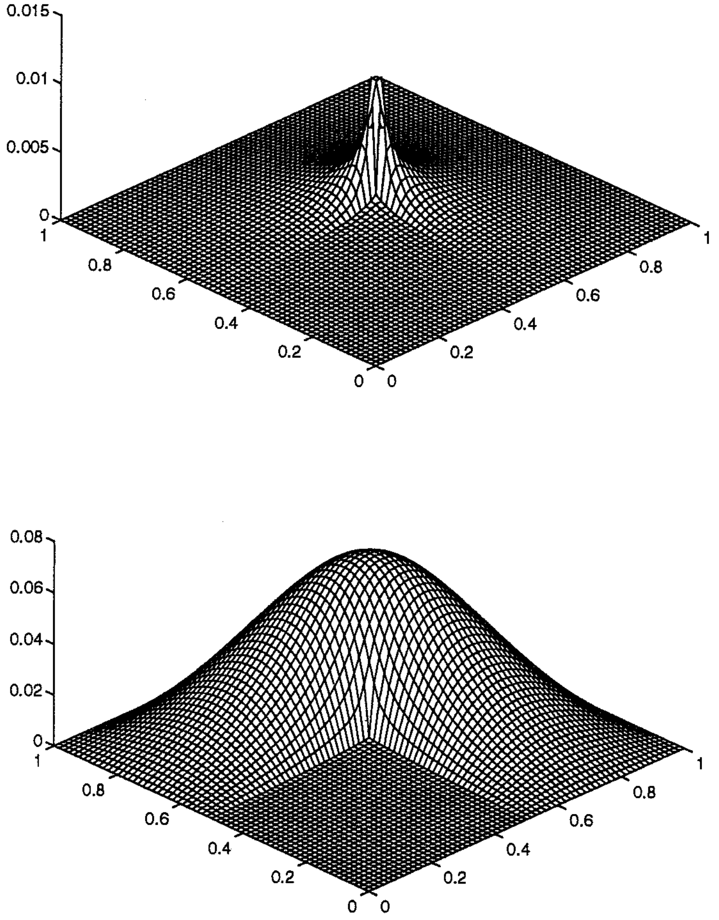


Figure 2: Example 2: Error in the pressure p for the weighted functional G_w^h (top) and the unmodified functional G^h (bottom)

REFERENCES

[1] R. E. Alcouffe, A. Brandt, J. J. E. Dendy, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.*, 2:430–454, 1981.

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

[3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer, New York, 1994.

[4] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. F. McCormick. First-order system least squares for second-order partial differential equations: Part I. *SIAM J. Numer. Anal.*, 31:1785–1799, 1994.

[5] Z. Cai, T. A. Manteuffel, and S. F. McCormick. First-order system least squares for second-order partial differential equations: Part II. *SIAM J. Numer. Anal.*, 1995. To Appear.

[6] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer, New York, 1986.

[7] M. J. Holst. *Multilevel Methods for the Poisson-Boltzmann Equation.* Ph.D. thesis, University of Illinois at Urbana-Champaign, 1993.

[8] O. E. Lafe, J. S. Montes, A. H. D. Cheng, J. A. Liggett, and P. L.-F. Liu. Singularities in Darcy flow through porous media. *J. ASCE Hydr. Div.*, 106:977–997, 1980.

[9] T. A. Manteuffel, S. F. McCormick, and G. Starke. Analysis of first-order system least-squares for elliptic problems with discontinuous coefficients. In Preparation.

[10] A. I. Pehlivanov and G. F. Carey. Error estimates for least-squares mixed finite elements. *RAIRO MMNA*, 28:499–516, 1994.

[11] A. I. Pehlivanov, G. F. Carey, and R. D. Lazarov. Least-squares mixed finite elements for second-order elliptic problems. *SIAM J. Numer. Anal.*, 31:1368–1377, 1994.

[12] T. F. Russell and M. F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. In R. E. Ewing, editor, *The Mathematics of Reservoir Simulation*, chapter II. SIAM, 1983.

[13] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* 2nd edition. Springer, New York, 1993.

[14] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

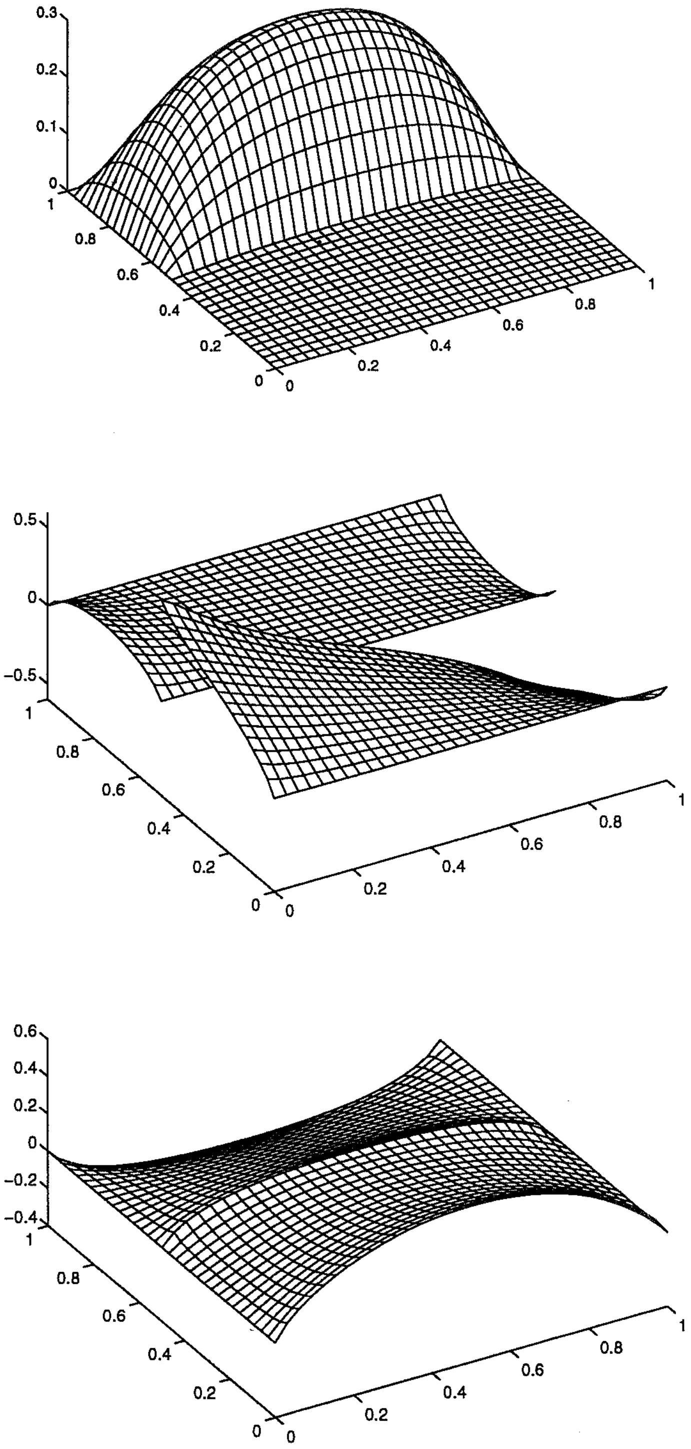


Figure 3: Example 1: Pressure p (top) and flux components u_1 and u_2

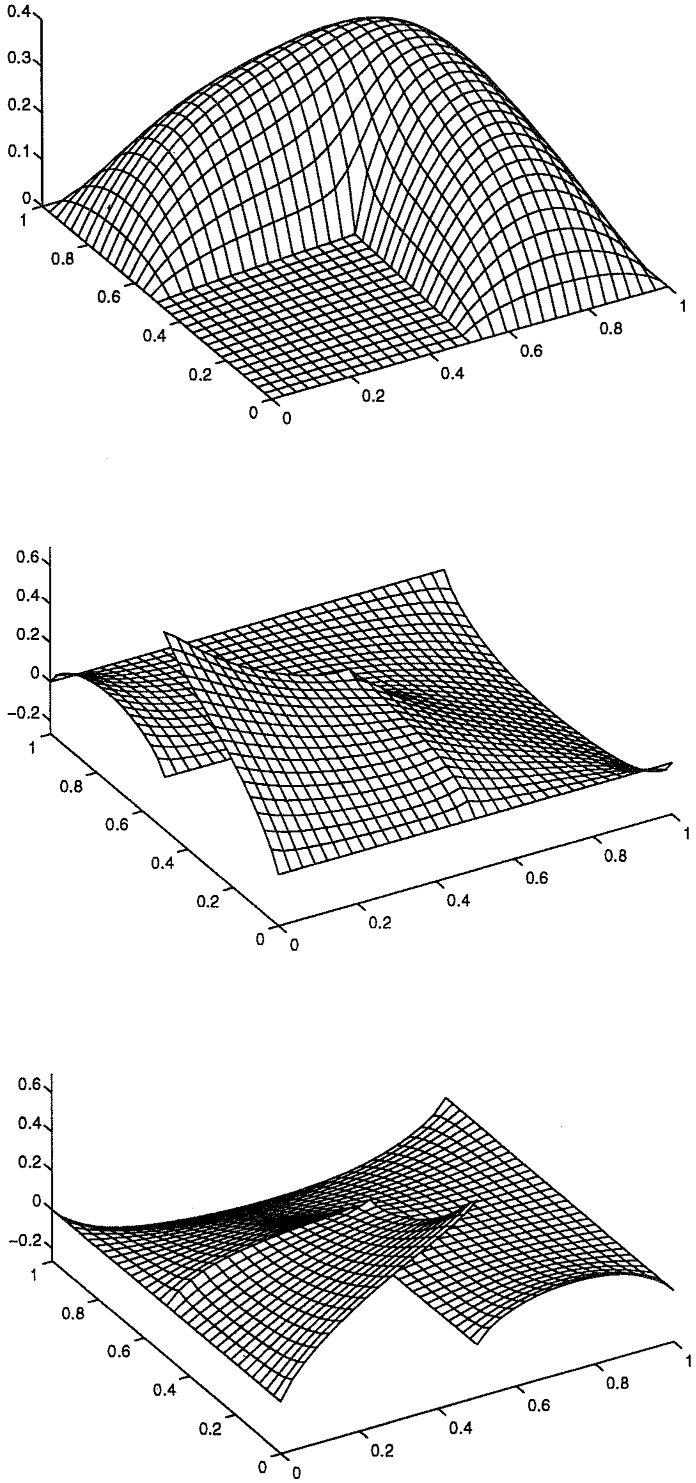


Figure 4: Example 2: Pressure p (top) and flux components u_1 and u_2

ON DGS RELAXATION: THE STOKES PROBLEM*

A. J. Meir
Department of Mathematics
Auburn University, AL

ABSTRACT

Multigrid methods have proven to be efficient methods for solving partial differential equations (especially those of elliptic type). There is also growing experience with multigrid solvers for fluids problems, e.g., the Stokes and Navier-Stokes equations (using both finite element and finite difference discretizations).

It is also well known that at the heart of any multigrid method is the smoother. In this work we look at a smoother introduced by Brandt and Dinar (DGS relaxation), and we examine some of its properties and consider some possible modifications to it. It is well known that multigrid performance using DGS relaxation is sensitive to the treatment of boundaries; this issue is addressed.

INTRODUCTION

Multigrid methods have proven to be efficient methods for solving partial differential equations (especially those of elliptic type). There is also growing experience with multigrid solvers for fluids problems, e.g., the Stokes and Navier-Stokes equations (using both finite element and finite difference discretizations. (See, e.g., [1]–[13] and the references therein.)

It is also well known that at the heart of any multigrid method is the smoother. In this work we look at a smoother (DGS relaxation; distributed Gauss-Seidel relaxation) introduced in [2] and [3], as it applies to the Stokes problem. We examine some of its properties and consider some possible modifications to it.

We consider the well-known Stokes equations; these equations, which model flows with small velocities (creeping flows), may be viewed as a linear version of the Navier-Stokes equations (which describe the flow of an incompressible, viscous fluid). The

*This work was supported in part by a contract from American Computing, Inc.

following analysis extends to the (nonlinear) Navier-Stokes equations and is the subject of a forthcoming paper.

The Stokes equations in Ω are, where Ω is a bounded domain in $\mathbb{R}^3$ (we assume the domain is three-dimensional; obviously, the following results hold equally well for two-dimensional domains),

$$-\Delta\mathbf{u} + \nabla p = \mathbf{f} \tag{1}$$

and

$$\nabla \cdot \mathbf{u} = 0\,. \tag{2}$$

On $\partial\Omega$ (the boundary of Ω),

$$\mathbf{u}|_{\partial\Omega} = \mathbf{g}\,. \tag{3}$$

Here $\mathbf{u}$ and p are the velocity and pressure, respectively (the unknowns). Given are the body force $\mathbf{f}$ and the boundary condition $\mathbf{g}$.

There exists a large body of work which deals with the analysis and the development of various approximation methods of solutions for this system of equations. (See, e.g., [14]–[17] and the references cited therein.) Here we propose yet another such method which is based on a reformulation of the equations (suggested by DGS relaxation).

Remark 1. *It is well known (see [15] and [16]) that given* $\mathbf{f} \in (H^1(\Omega)^3)^*$ *and* $\mathbf{g} \in H^{1/2}(\partial\Omega)^3$ *with* $\int_{\partial\Omega} \mathbf{g}\cdot\mathbf{n}\,ds = 0$ *the Stokes equations (1)–(3) have a unique solution* $(\mathbf{u}, p) \in H^1(\Omega)^3 \times L_0^2(\Omega)$.

Throughout the paper we assume that Ω is a bounded, simply-connected domain in $\mathbb{R}^3$ which is of class $C^{1,1}$ or is a convex polyhedron. (See [16] or [18].) The boundary of the domain is denoted $\partial\Omega$ and $\mathbf{n}$ is the unit outward-pointing normal vector to Ω. Here and in the sequel $H^s(\Omega)$ (s a positive integer) is the usual $L^2(\Omega)$-based Sobolev space, $H^{1/2}(\partial\Omega)$ is the trace space of $H^1(\Omega)$, and $H^{-1/2}(\partial\Omega)$ is its dual. (See [18].) Also,

$$L_0^2(\Omega) = \left\{p \in L^2(\Omega) : \int_\Omega p\,dx = 0\right\}$$

(i.e., it is the subspace of L^2-functions which have zero mean; see [16] and [17]). We also introduce the following subspaces of $H^{-1/2}(\partial\Omega)^3$ and $H^1(\Omega)^3$ (see [19] and [16]):

$$H_n^{-1/2}(\Omega)^3 := \left\{\mathbf{t} \in H^{-1/2}(\partial\Omega)^3 : \mathbf{t}\cdot\mathbf{n} = 0\right\}$$

and

$$H_n^1(\Omega)^3 := \left\{\Psi \in H^1(\Omega)^3 : \Psi\cdot\mathbf{n}|_{\partial\Omega} = 0\right\}.$$

On $H_0^1(\Omega)^3$ (the space of functions with zero trace on the boundary) and on $H_n^1(\Omega)^3$,

$$\left(\|\nabla\times(\,\cdot\,)\|_0^2 + \|\nabla\cdot(\,\cdot\,)\|_0^2\right)^{1/2}$$

is a norm equivalent to the H^1-norm (due to the existence of a Poincaré-type inequality for domains such as those discussed above; see, e.g., [16]). Here $\|\cdot\|_s$ denotes the H^s-norm ($s = 0$ for L^2).

The Stokes equations can be formally written as the system

$$L\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix}=\begin{bmatrix}-\Delta & \nabla\\ -\nabla\cdot & 0\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix}=\begin{bmatrix}\mathbf{f}\\ 0\end{bmatrix},$$

$$\mathbf{u}|_{\partial\Omega}=\mathbf{g}\,.$$

DGS relaxation may be viewed as Gauss-Seidel relaxation on a right preconditioned system or Gauss-Seidel relaxation on an equation with transformed variables. The change of variables (up to a sign change) as described in [2] and [3] (also in [13]) is given as

$$LM\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{p}\end{bmatrix}=\begin{bmatrix}-\Delta & 0\\ -\nabla\cdot & -\Delta\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{p}\end{bmatrix}=\begin{bmatrix}\mathbf{f}\\ 0\end{bmatrix}.$$

It is easily seen that the (so called) distribution matrix M (the right preconditioner) is

$$M=\begin{bmatrix}I & \nabla\\ 0 & \Delta\end{bmatrix}.$$

Formally, M^{-1}, the inverse change of variables, is given by

$$M^{-1}=\begin{bmatrix}I & -\nabla\Delta^{-1}\\ 0 & \Delta^{-1}\end{bmatrix}.$$

So the change of variables is given by

$$\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix}=\begin{bmatrix}I & \nabla\\ 0 & \Delta\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{p}\end{bmatrix}\quad\text{or}\quad\begin{bmatrix}\tilde{\mathbf{u}}\\ \tilde{p}\end{bmatrix}=\begin{bmatrix}I & -\nabla\Delta^{-1}\\ 0 & \Delta^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix}.$$

Thus we end up with the equations

$$-\Delta\tilde{\mathbf{u}}=\mathbf{f}$$

and

$$-\Delta\tilde{p}=\nabla\cdot\tilde{\mathbf{u}}\,.$$

An obvious obstacle in this approach is the lack of boundary conditions on $\tilde{\mathbf{u}}=\mathbf{u}-\nabla\Delta^{-1}p=\mathbf{u}-\nabla\tilde{p}$ and on $\tilde{p}=\Delta^{-1}p$. Obviously we cannot specify $\nabla\tilde{p}$ on the boundary (one would like to do that since $\mathbf{u}|_{\partial\Omega}=\mathbf{g}$ is given), since this would result in an overdetermined system for $\tilde{p}$. Note that even if a boundary condition for $\tilde{p}$ were derived and we were to derive a boundary condition for $\tilde{\mathbf{u}}$, this boundary condition for $\tilde{\mathbf{u}}$ would involve $\tilde{p}$ (namely, $\nabla\tilde{p}$). Thus we would end up with a system of equations that are coupled through the boundary conditions. (See [4].)

Thus it is proposed (in [2] and [3]) that this system be solved iteratively (with no mention of the boundary conditions to be used); that is, we perform a Gauss-Seidel step on the transformed system and then perform the inverse change of variables. In practice we only work with the original variables (the new variables are introduced only to describe the method). In fact, some ad hoc modifications to the method are proposed in [13]; these improve the method in the presence of boundaries.

An obvious question is whether other changes of variables may yield a similar iteration scheme. (See [5].) The most obvious change of variables that comes to mind will avoid forming the Laplacian and inverse Laplacian in the equation for the pressure; it will therefore be given by the following distribution matrix:

$$M = \begin{bmatrix} I & \Delta^{-1}\nabla \\ 0 & I \end{bmatrix} .$$

Formally, the inverse of the distribution matrix is

$$M^{-1} = \begin{bmatrix} I & -\Delta^{-1}\nabla \\ 0 & I \end{bmatrix} .$$

Now

$$LM = \begin{bmatrix} -\Delta & 0 \\ -\nabla\cdot & -I \end{bmatrix} ,$$

so the change of variables is given by

$$\begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} I & \Delta^{-1}\nabla \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{p} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} I & -\Delta^{-1}\nabla \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} .$$

This change of variables will yield a relaxation method which we call MDGS (modified DGS) relaxation.

Thus we end up with the equations

$$-\Delta\hat{\mathbf{u}} = \mathbf{f}$$

and

$$-\hat{p} = \nabla \cdot \hat{\mathbf{u}} .$$

An obvious advantage of this method is that there are no additional boundary conditions which must be imposed (or, more precisely, we may impose the boundary condition $\hat{\mathbf{u}}|_{\partial\Omega} = \mathbf{g}$, and no boundary condition is needed for $\hat{p}$). A drawback of the method is that it is more complicated (since the change of variables now involves an inverse Laplacian, although this can be approximated locally). This alternative is very similar to an iteration for Uzawa's method; see [6], [20], and [21]. (See also [14] and [16].)

We abandon, for the time being, any further discussion of DGS (and MDGS) relaxation and consider a related alternate formulation of the Stokes problem.

ALTERNATE FORMULATION

We consider the following formulation for the Stokes problem:

$$-\Delta\mathbf{v} = \mathbf{f} , \tag{4}$$

$$\nabla \cdot \mathbf{\Phi} = -\nabla \cdot \mathbf{v}\,, \tag{5}$$

and

$$\nabla \times \mathbf{\Phi} = 0\,. \tag{6}$$

With boundary conditions

$$\mathbf{v}|_{\partial\Omega} + \mathbf{\Phi}|_{\partial\Omega} = \mathbf{g} \tag{7}$$

and

$$\mathbf{\Phi} \cdot \mathbf{n}|_{\partial\Omega} = 0\,. \tag{8}$$

An alternate formulation with boundary condition $\mathbf{\Phi} \times \mathbf{n}|_{\partial\Omega} = 0$ (instead of (8)) may be treated as well; details will appear in a forthcoming paper.

This formulation is equivalent to the Stokes equations when we set the velocity

$$\mathbf{u} = \mathbf{v} + \mathbf{\Phi} \tag{9}$$

and the pressure

$$p = \nabla \cdot \mathbf{\Phi}\,. \tag{10}$$

Note that if (8) is satisfied then $\int_\Omega \nabla \cdot \mathbf{\Phi}\, dx = 0$, and we may in fact (due to (5)) set $p = -\nabla \cdot \mathbf{v}$.

Since $\mathbf{\Phi}$ satisfies

$$\nabla \cdot \mathbf{\Phi} = -\nabla \cdot \mathbf{v}\,, \tag{11}$$

$$\nabla \times \mathbf{\Phi} = 0\,, \tag{12}$$

and

$$\mathbf{\Phi} \cdot \mathbf{n}|_{\partial\Omega} = 0\,, \tag{13}$$

there exists ϕ such that $\mathbf{\Phi} = \nabla\phi$; moreover, ϕ is characterized as the solution of

$$-\Delta\phi = -\nabla \cdot \mathbf{\Phi} = \nabla \cdot \mathbf{v} \tag{14}$$

and

$$\nabla\phi \cdot \mathbf{n}|_{\partial\Omega} = 0\,. \tag{15}$$

Because $\mathbf{\Phi} = \nabla\phi$, the fact that ϕ (the solution of (14) and (15)) is unique only up to an additive constant does not cause any difficulties.

In light of the above, one may replace (4)–(8) by

$$-\Delta\mathbf{v} = \mathbf{f}\,, \tag{16}$$

$$-\Delta\phi = \nabla \cdot \mathbf{v}\,, \tag{17}$$

$$\mathbf{v}|_{\partial\Omega} + \nabla\phi|_{\partial\Omega} = \mathbf{g}\,, \tag{18}$$

and

$$\nabla\phi \cdot \mathbf{n}|_{\partial\Omega} = 0\,. \tag{19}$$

The relationship of this formulation to DGS and MDGS is now patently clear if we identify

$$\hat{\mathbf{u}} = \tilde{\mathbf{u}} = \mathbf{v} \quad \text{and} \quad \hat{p} = \Delta\tilde{p} = -\nabla \cdot \mathbf{v} = \nabla \cdot \mathbf{\Phi} = \Delta\phi\,.$$

The advantage of this point of view is the availability of boundary conditions for the various unknowns. A difficulty in this approach is the fact that the equations are coupled through the boundary conditions; this situation is unavoidable, however (as observed earlier). We also have the following theorem:

Theorem 2. *The formulation (1)–(3) is equivalent to the formulation (4)–(8) and to the formulation (16)–(19).*

Proof: If $(\mathbf{u}, p) \in H^1(\Omega)^3 \times L_0^2(\Omega)$ is a solution of (1)–(3) then let $\mathbf{\Phi}$ be the unique solution of

$$\nabla \cdot \mathbf{\Phi} = p\,,$$

$$\nabla \times \mathbf{\Phi} = 0\,,$$

and

$$\mathbf{\Phi} \cdot \mathbf{n}|_{\partial\Omega} = 0\,.$$

Note that $\nabla \cdot \mathbf{u} = 0$ and $\Delta\mathbf{\Phi} = \nabla\nabla \cdot \mathbf{\Phi}$ (due to the fact that $-\Delta\mathbf{\Phi} = \nabla \times \nabla \times \mathbf{\Phi} - \nabla\nabla \cdot \mathbf{\Phi}$ and $\nabla \times \mathbf{\Phi} = 0$); thus, $\Delta\mathbf{\Phi} = \nabla p$. Setting

$$\mathbf{v} = \mathbf{u} - \mathbf{\Phi}\,,$$

it is easily seen that $(\mathbf{v}, \mathbf{\Phi})$ satisfies (4)–(8). Conversely, if $(\mathbf{v}, \mathbf{\Phi})$ satisfies (4)–(8), then set

$$\mathbf{u} = \mathbf{v} + \mathbf{\Phi}$$

and

$$p = \nabla \cdot \mathbf{\Phi}\,.$$

Recall $\Delta\mathbf{\Phi} = \nabla p$; clearly $(\mathbf{u}, p)$ satisfies (1)–(3).

It is well known that (5)–(6) and (8) are equivalent to (14) and (15), with the identification $\mathbf{\Phi} = \nabla\phi$. (See, e.g., [16].) To complete the proof we observe the following: if $(\mathbf{u}, p)$ satisfies the Stokes equations and if we set

$$-\Delta\phi = -p\,,$$

$$\nabla\phi \cdot \mathbf{n}|_{\partial\Omega} = 0\,,$$

and

$$\mathbf{v} = \mathbf{u} - \nabla\phi\,,$$

then $(\mathbf{v}, \phi)$ so defined satisfies equations (16)–(19). Conversely, if $(\mathbf{v}, \phi)$ satisfies equations (16)–(19), set

$$\mathbf{u} = \mathbf{v} + \nabla\phi$$

and

$$p = \Delta\phi\,,$$

then $(\mathbf{u}, p)$ satisfies the Stokes problem (equations (1)–(3)). □

Consider the following weak formulation: find $\mathbf{v}$, $\mathbf{s}$, and $\mathbf{\Phi}$ such that

$$\mathbf{v} \in H^1(\Omega)^3 \quad \text{with} \quad \mathbf{v} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{g} \cdot \mathbf{n}\,, \quad \mathbf{s} \in H_n^{-1/2}(\partial\Omega)^3\,, \quad \mathbf{\Phi} \in H_n^1(\Omega)^3\,, \tag{20}$$

$$\begin{aligned}&\int_\Omega \{\nabla \times \mathbf{v} \cdot \nabla \times \mathbf{w} + \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{w} + \nabla \times \mathbf{\Phi} \cdot \nabla \times \mathbf{\Psi} + \nabla \cdot \mathbf{\Phi} \nabla \cdot \mathbf{\Psi}\}\, dx \\ &\quad + \int_\Omega \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{\Psi}\, dx + \langle \mathbf{s}, \mathbf{w} \rangle_{\partial\Omega} = \langle \mathbf{f}, \mathbf{w} \rangle_\Omega \qquad \forall \mathbf{w} \in H_n^1(\Omega)^3\,, \mathbf{\Psi} \in H_n^1(\Omega)^3\,,\end{aligned} \tag{21}$$

and

$$\langle \mathbf{t}, \mathbf{v} + \mathbf{\Phi} \rangle_{\partial\Omega} = \langle \mathbf{t}, \mathbf{g} \rangle_{\partial\Omega} \qquad \forall \mathbf{t} \in H_n^{-1/2}(\partial\Omega)^3\,. \tag{22}$$

Here $\langle \cdot, \cdot \rangle_\Omega$ and $\langle \cdot, \cdot \rangle_{\partial\Omega}$ denote the duality pairing of $H^1(\Omega)^3$ and $(H^1(\Omega)^3)^*$ and of $H^{1/2}(\partial\Omega)^3$ and $H^{-1/2}(\partial\Omega)^3$, respectively. Or equivalently, consider the following weak formulation: find $\mathbf{v}$, $\mathbf{s}$, and ϕ such that

$$\begin{aligned}\mathbf{v} \in H^1(\Omega)^3 \quad \text{with} \quad \mathbf{v} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{g} \cdot \mathbf{n}\,, \quad \mathbf{s} \in H_n^{-1/2}(\partial\Omega)^3\,, \\ \phi \in H^2(\Omega) \quad \text{with} \quad \nabla\phi \in H_n^1(\Omega)^3\,,\end{aligned} \tag{23}$$

$$\begin{aligned}&\int_\Omega \{\nabla \times \mathbf{v} \cdot \nabla \times \mathbf{w} + \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{w} + \Delta\phi\Delta\psi\}\, dx + \int_\Omega \nabla \cdot \mathbf{v} \Delta\psi\, dx + \langle \mathbf{s}, \mathbf{w} \rangle_{\partial\Omega} \\ &\quad = \langle \mathbf{f}, \mathbf{w} \rangle_\Omega \qquad \forall \mathbf{w} \in H_n^1(\Omega)^3\,, \psi \in H^2(\Omega) \quad \text{with} \quad \nabla\psi \in H_n^1(\Omega)^3\,,\end{aligned} \tag{24}$$

and

$$\langle \mathbf{t}, \mathbf{v} + \nabla\phi \rangle_{\partial\Omega} = \langle \mathbf{t}, \mathbf{g} \rangle_{\partial\Omega} \qquad \forall \mathbf{t} \in H_n^{-1/2}(\partial\Omega)^3\,. \tag{25}$$

Theorem 3. *Equations (20)–(22) and (23)–(25) are weak formulations for (4)–(8) and (16)–(19), respectively.*

Proof: Setting $\mathbf{\Psi} = 0$ and restricting $\mathbf{w} \in H_0^1(\Omega)^3$ in (21) we get that

$$\int_\Omega \{\nabla \times \mathbf{v} \cdot \nabla \times \mathbf{w} + \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{w}\}\, dx = \langle \mathbf{f}, \mathbf{w} \rangle_\Omega\,,$$

which implies that

$$-\Delta\mathbf{v} = \mathbf{f}$$

in $H^{-1}(\Omega)^3$; letting $\mathbf{w}$ be an arbitrary element of $H_n^1(\Omega)^3$ we get that

$$\mathbf{s} = -\nabla \times \mathbf{v} \times \mathbf{n}|_{\partial\Omega}$$

in $H^{-1/2}(\partial\Omega)^3$. Now setting $\mathbf{w} = 0$ and setting $\mathbf{\Psi}$ to be the solution of

$$\nabla \cdot \mathbf{\Psi} = \nabla \cdot \mathbf{\Phi} + \nabla \cdot \mathbf{v}\,,$$

$$\nabla \times \mathbf{\Psi} = 0\,,$$

and

$$\mathbf{\Psi} \cdot \mathbf{n}|_{\partial\Omega} = 0\,,$$

we get that

$$\nabla \cdot \mathbf{\Phi} = -\nabla \cdot \mathbf{v}$$

in $L^2(\Omega)$. Letting $\mathbf{\Psi}$ be an arbitrary element of $H^1_n(\Omega)^3$ we get that

$$\nabla \times \mathbf{\Phi} = 0$$

in $L^2(\Omega)^3$. Finally from (20) and (22) we obtain (7). The proof for the formulation (23)–(25) proceeds similarly. □

For notational convenience, define

$$\begin{aligned}
A((\mathbf{v}, \mathbf{\Phi}), (\mathbf{w}, \mathbf{\Psi})) &:= \int_\Omega \{\nabla \times \mathbf{v} \cdot \nabla \times \mathbf{w} + \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{w} + \nabla \times \mathbf{\Phi} \cdot \nabla \times \mathbf{\Psi}\}\, dx \\
&\quad + \int_\Omega \nabla \cdot \mathbf{\Phi} \nabla \cdot \mathbf{\Psi}\, dx + \int_\Omega \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{\Psi}\, dx\,, \\
B(\mathbf{s}, (\mathbf{w}, \mathbf{\Psi})) &:= \langle \mathbf{s}, \mathbf{w} \rangle_{\partial\Omega}\,, \\
D(\mathbf{t}, (\mathbf{v}, \mathbf{\Phi})) &:= \langle \mathbf{t}, \mathbf{v} + \mathbf{\Phi} \rangle_{\partial\Omega}\,, \\
F((\mathbf{w}, \mathbf{\Psi})) &:= \langle \mathbf{f}, \mathbf{w} \rangle_{\Omega}\,, \\
G(\mathbf{t}) &:= \langle \mathbf{t}, \mathbf{g} \rangle_{\partial\Omega}\,,
\end{aligned}$$

and

$$\begin{aligned}
a((\mathbf{v}, \phi), (\mathbf{w}, \psi)) &:= \int_\Omega \{\nabla \times \mathbf{v} \cdot \nabla \times \mathbf{w} + \nabla \cdot \mathbf{v} \nabla \cdot \mathbf{w} + \Delta\phi \Delta\psi\}\, dx + \int_\Omega \nabla \cdot \mathbf{v} \Delta\psi\, dx\,, \\
b(\mathbf{s}, (\mathbf{w}, \psi)) &:= \langle \mathbf{s}, \mathbf{w} \rangle_{\partial\Omega}\,, \\
d(\mathbf{t}, (\mathbf{v}, \phi)) &:= \langle \mathbf{t}, \mathbf{v} + \nabla\phi \rangle_{\partial\Omega}\,, \\
f((\mathbf{w}, \psi)) &:= \langle \mathbf{f}, \mathbf{w} \rangle_{\Omega}\,, \\
g(\mathbf{t}) &:= \langle \mathbf{t}, \mathbf{g} \rangle_{\partial\Omega}\,.
\end{aligned}$$

We denote

$$\mathcal{H} := H^1(\Omega)^3 \times H^1_n(\Omega)^3$$

and

$$\mathcal{H}_n := H^1_n(\Omega)^3 \times H^1_n(\Omega)^3\,.$$

On these spaces we use the usual product norm.

With this notation we may write the weak formulations as follows: find $\mathbf{v}$, $\mathbf{s}$, and $\mathbf{\Phi}$ such that

$$\mathbf{v} \in H^1(\Omega)^3 \quad \text{with} \quad \mathbf{v} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{g} \cdot \mathbf{n}\,, \quad \mathbf{s} \in H_n^{-1/2}(\partial\Omega)^3\,, \quad \mathbf{\Phi} \in H^1_n(\Omega)^3\,, \tag{26}$$

$$A((\mathbf{v}, \mathbf{\Phi}), (\mathbf{w}, \mathbf{\Psi})) + B(\mathbf{s}, (\mathbf{w}, \mathbf{\Psi})) = F((\mathbf{w}, \mathbf{\Psi})) \qquad \forall (\mathbf{w}, \mathbf{\Psi}) \in \mathcal{H}_n\,, \tag{27}$$

and

$$D(\mathbf{t}, (\mathbf{v}, \mathbf{\Phi})) = G(\mathbf{t}) \qquad \forall \mathbf{t} \in H_n^{-1/2}(\partial\Omega)^3 . \tag{28}$$

Equivalently, find $\mathbf{v}$, $\mathbf{s}$, and ϕ such that

$$\begin{gathered} \mathbf{v} \in H^1(\Omega)^3 \quad \text{with} \quad \mathbf{v} \cdot \mathbf{n}|_{\partial\Omega} = \mathbf{g} \cdot \mathbf{n} , \quad \mathbf{s} \in H_n^{-1/2}(\partial\Omega)^3 , \\ \phi \in H^2(\Omega) \quad \text{with} \quad \nabla\phi \in H_n^1(\Omega)^3 , \end{gathered} \tag{29}$$

$$\begin{gathered} a((\mathbf{v}, \phi), (\mathbf{w}, \psi)) + b(\mathbf{s}, (\mathbf{w}, \psi)) = f((\mathbf{w}, \psi)) \\ \forall \mathbf{w} \in H_n^1(\Omega)^3 , \psi \in H^2(\Omega) \quad \text{with} \quad \nabla\psi \in H_n^1(\Omega)^3 , \end{gathered} \tag{30}$$

and

$$d(\mathbf{t}, (\mathbf{v}, \phi)) = g(\mathbf{t}) \qquad \forall \mathbf{t} \in H_n^{-1/2}(\partial\Omega)^3 . \tag{31}$$

Note that this weak formulation falls into the class of generalized saddle point problems of the type considered in [22]. (See also [14] and [23].)

Lemma 4. *The forms $A(\cdot,\cdot)$, $B(\cdot,\cdot)$, $D(\cdot,\cdot)$, $F(\cdot)$, and $G(\cdot)$ are continuous; that is, positive constants λ_A, λ_B, λ_D, λ_F, and λ_G exist such that*

$$|A((\mathbf{v}, \mathbf{\Phi}), (\mathbf{w}, \mathbf{\Psi}))| \le \lambda_A \|(\mathbf{v}, \mathbf{\Phi})\|_{\mathcal{H}} \|(\mathbf{w}, \mathbf{\Psi})\|_{\mathcal{H}} , \tag{32}$$

$$|B(\mathbf{s}, (\mathbf{w}, \mathbf{\Psi}))| \le \lambda_B \|\mathbf{s}\|_{-1/2} \|(\mathbf{w}, \mathbf{\Psi})\|_{\mathcal{H}} , \tag{33}$$

$$|D(\mathbf{t}, (\mathbf{v}, \mathbf{\Phi}))| \le \lambda_D \|\mathbf{t}\|_{-1/2} \|(\mathbf{v}, \mathbf{\Phi})\|_{\mathcal{H}} , \tag{34}$$

$$|F((\mathbf{w}, \mathbf{\Psi}))| \le \lambda_F \|(\mathbf{w}, \mathbf{\Psi})\|_{\mathcal{H}} , \tag{35}$$

and

$$|G(\mathbf{t})| \le \lambda_G \|\mathbf{t}\|_{-1/2} . \tag{36}$$

Proof: The proof is an easy consequence of Hölder's inequality and the definition of the forms. □

Define

$$\mathcal{K}_B := \{(\mathbf{w}, \mathbf{\Psi}) \in \mathcal{H}_n : B(\mathbf{s}, (\mathbf{w}, \mathbf{\Psi})) = 0 \quad \forall \mathbf{s} \in H_n^{-1/2}(\partial\Omega)^3\} ,$$

and

$$\mathcal{K}_D := \{(\mathbf{v}, \mathbf{\Phi}) \in \mathcal{H}_n : D(\mathbf{t}, (\mathbf{v}, \mathbf{\Phi})) = 0 \quad \forall \mathbf{t} \in H_n^{-1/2}(\partial\Omega)^3\} .$$

Lemma 5. *The forms $A(\cdot,\cdot)$, $B(\cdot,\cdot)$, and $D(\cdot,\cdot)$ satisfy some inf-sup conditions; in particular, positive constants α, β, and δ exist such that*

$$\inf_{(\mathbf{w},\mathbf{\Psi}) \in \mathcal{K}_B} \sup_{(\mathbf{v},\mathbf{\Phi}) \in \mathcal{K}_D} \frac{A((\mathbf{v}, \mathbf{\Phi}), (\mathbf{w}, \mathbf{\Psi}))}{\|(\mathbf{v}, \mathbf{\Phi})\|_{\mathcal{H}} \|(\mathbf{w}, \mathbf{\Psi})\|_{\mathcal{H}}} \ge \alpha , \tag{37}$$

$$\inf_{(\mathbf{v},\mathbf{\Phi}) \in \mathcal{K}_D \setminus \{0\}} \sup_{(\mathbf{w},\mathbf{\Psi}) \in \mathcal{K}_B} A((\mathbf{v}, \mathbf{\Phi}), (\mathbf{w}, \mathbf{\Psi})) > 0 , \tag{38}$$

$$\inf_{\mathbf{s}\in H_n^{-1/2}(\partial\Omega)} \sup_{(\mathbf{w},\mathbf{\Psi})\in\mathcal{H}_n} \frac{B(\mathbf{s},(\mathbf{w},\mathbf{\Psi}))}{\|\mathbf{s}\|_{-1/2}\|(\mathbf{w},\mathbf{\Psi})\|_{\mathcal{H}}} \geq \beta\,, \tag{39}$$

and

$$\inf_{\mathbf{t}\in H_n^{-1/2}(\partial\Omega)} \sup_{(\mathbf{v},\mathbf{\Phi})\in\mathcal{H}_n} \frac{D(\mathbf{t},(\mathbf{v},\mathbf{\Phi}))}{\|\mathbf{t}\|_{-1/2}\|(\mathbf{v},\mathbf{\Phi})\|_{\mathcal{H}}} \geq \delta\,. \tag{40}$$

Proof: The first condition (inequality (37)) follows from the observations that given $(\mathbf{w},\mathbf{\Psi}) \in \mathcal{K}_B$, setting $\mathbf{v} = \mathbf{w} - \mathbf{\Psi}$ and $\mathbf{\Phi} = \mathbf{\Psi}$ guarantees that $(\mathbf{v},\mathbf{\Phi}) \in \mathcal{K}_D$, that a positive constant c exists so that $\|(\mathbf{v},\mathbf{\Phi})\|_{\mathcal{H}} < c\|(\mathbf{w},\mathbf{\Psi})\|_{\mathcal{H}}$, and that

$$A((\mathbf{v},\mathbf{\Phi}),(\mathbf{w},\mathbf{\Psi})) \geq \frac{1}{2}\|(\mathbf{w},\mathbf{\Psi})\|_{\mathcal{H}}^2\,.$$

Given $(\mathbf{v},\mathbf{\Phi}) \in \mathcal{K}_D\setminus\{0\}$, set $\mathbf{w} = \mathbf{v}+\mathbf{\Phi}$ and $\mathbf{\Psi} = \mathbf{\Phi}$; then, $(\mathbf{w},\mathbf{\Psi}) \in \mathcal{K}_B$; moreover, it is easily seen that

$$A((\mathbf{v},\mathbf{\Phi}),(\mathbf{w},\mathbf{\Psi})) \geq \frac{1}{2}\left(\|\nabla\times\mathbf{v}\|_0^2 + \|\nabla\times\mathbf{\Phi}\|_0^2\right) + \|\nabla\cdot(\mathbf{v}+\mathbf{\Phi})\|_0^2\,.$$

Now if $\frac{1}{2}(\|\nabla\times\mathbf{v}\|_0^2 + \|\nabla\times\mathbf{\Phi}\|_0^2) + \|\nabla\cdot(\mathbf{v}+\mathbf{\Phi})\|_0^2 > 0$, then (38) holds. If this is not the case (i.e., if $\frac{1}{2}(\|\nabla\times\mathbf{v}\|_0^2 + \|\nabla\times\mathbf{\Phi}\|_0^2) + \|\nabla\cdot(\mathbf{v}+\mathbf{\Phi})\|_0^2 = 0$), it easily follows that $\mathbf{v}+\mathbf{\Phi} = 0$, and, because $(\mathbf{v},\mathbf{\Phi}) \neq 0$, then $\nabla\cdot\mathbf{v} \neq 0$. In this case we know (see [16]) that a $\mathbf{w} \in H_0^1(\Omega)$ exists with $\nabla\cdot\mathbf{w} = \nabla\cdot\mathbf{v}$; setting $\mathbf{\Psi} = 0$, we get that

$$A((\mathbf{v},\mathbf{\Phi}),(\mathbf{w},\mathbf{\Psi})) \geq \|\nabla\cdot\mathbf{v}\|_0^2$$

and conclude that the second condition holds.

The third and fourth conditions (inequalities (39) and (40)) may be shown using the methods used in [24] to prove a similar inf-sup condition. □

Theorem 6. *The weak problem (26)–(28) has a unique solution.*

Proof: This is a result of Lemma 4, Lemma 5, and the abstract theory detailed in [22] and [23]. □

It is an easy exercise to state, for (29)–(31), results analogous to those stated in Lemma 4, Lemma 5, and Theorem 6. (Details will be given in a forthcoming paper.)

DISCUSSION

We point out that since the weak form of the problem falls into the class of generalized saddle point problems introduced in [22] (see also [14] and [23]), one may carry out finite element analysis for this problem in that framework. Such analysis yields existence and uniqueness results for the discrete problem (approximate problem) and optimal error estimates for finite element approximation schemes based on these weak forms, provided that certain (discrete) inf-sup conditions hold. (Details will be given in a forthcoming paper.)

An advantage of this formulation over the primitive variable (velocity-pressure) formulation of the problem is the fact that it is relatively easy to construct finite element spaces which satisfy the necessary inf-sup conditions. In fact there is complete freedom in choosing the spaces for $\mathbf{v}$ and for $\mathbf{\Phi}$ (the spaces that approximate $H^1(\Omega)^3$ and $H^1_n(\Omega)^3$); in view of the error estimates it is reasonable to choose the same finite element space for both of these. Once these spaces have been chosen, we choose the space for $\mathbf{s}$ (the space approximating $H^{-1/2}_n(\partial\Omega)^3$) as the restriction to the boundary of elements of the previous spaces (i.e., the trace space of the discrete spaces approximating $H^1_n(\Omega)^3$). This choice for the discrete spaces guarantees that the necessary (discrete) inf-sup conditions are satisfied. Details and examples from computations will appear in a forthcoming paper.

Another question to be investigated is the implications for multigrid codes employing DGS relaxation. Can these results be used in order to construct better smoothers (particularly in the neighborhood of boundaries)? As stated earlier, the relationship between this formulation and DGS relaxation is

$$\tilde{\mathbf{u}} = \mathbf{v} \quad \text{and} \quad \Delta\tilde{p} = -\nabla\cdot\mathbf{v} = \nabla\cdot\mathbf{\Phi}\,,$$

but we also have that

$$\mathbf{u} = \tilde{\mathbf{u}} + \nabla\tilde{p} = \mathbf{v} + \mathbf{\Phi}\,.$$

Therefore it seems that when using DGS relaxation one alternative is to impose a homogeneous Neumann boundary condition on $\tilde{p}$ (when solving $-\Delta\tilde{p} = \nabla\cdot\tilde{\mathbf{u}}$) and the nonhomogeneous Dirichlet boundary condition $\mathbf{g}-\nabla\tilde{p}|_{\partial\Omega}$ on $\tilde{\mathbf{u}}$ (when solving $-\Delta\tilde{\mathbf{u}} = \mathbf{f}$).

Moreover it may prove advantageous to keep explicit track of $\tilde{u}$ and $\tilde{p}$ on the boundary and use their values in the iteration. This may yield better behavior of DGS relaxation in the presence of boundaries.

DGS relaxation (the change of variables described in [2] and [3]) is introduced in order to transform a saddle point problem into a problem which is definite. The fact that the new problem is still indefinite (a saddle point problem) is masked by the fact that the effects of the boundaries and boundary conditions have been neglected. Based on the previous analysis it is obvious that we are still faced with an indefinite problem. This must be taken into account when using this iterative scheme; one possible implication is that it may be advantageous to use an inexact Uzawa-type iteration to solve the problem.

REFERENCES

1. Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems, Math. Comp, vol. 31, no. 138, 1977, pp. 330–390.

2. Brandt, A.; and Dinar, N.: Multigrid Solutions to Elliptic Flow Problems, in Numerical Methods for Partial Differential Equations, S. V. Parter ed., Academic Press, New York, 1979.

3. Brandt, A.: Multigrid Techniques: 1984 Guide With Applications to Fluid Dynamics, The Weizmann Institute, Rehovot, 1984.

4. Fuchs, L.; and Zhao, H.-S.: Solutions of Three-Dimensional Viscous Incompressible Flows by a Multi-Grid Method, Int. J. Num. Meth. Fluids, vol. 4, 1984, pp. 539–555.

5. Linden, J.; Lonsdale, G.; Steckel, B.; and Stüben, K.: Multigrid for the Steady-State Incompressible Navier-Stokes Equations: A Survey, Arbeitspapiere der GMD 322, 1988.

6. Maitre, J. F.; Musy, F.; and Nigon, P.: A Fast Solver for the Stokes Equations Using Multigrid with a Uzawa Smoother, in Advances in Multigrid, D. Braess, W. Hackbusch, and U. Trottenberg eds., Friedr. Vieweg & Sohn, Braunschweig, 1985.

7. Niestegge, A.; Witsch, K.: Analysis of a Multigrid Stokes Solver, Appl. Math. Comput., vol. 35, 1990, pp. 291–303.

8. Verfurth, R.: A Combined Conjugate Gradient-Multigrid Algorithm for the Numerical Solution of the Stokes Problem, IMA J. Numer. Anal., vol. 4, 1984, pp. 441–455.

9. Verfurth, R.: A Multilevel Algorithm for Mixed Problems, SIAM J. Numer. Anal., vol. 21, no. 2, 1984, pp. 264–271.

10. Verfurth, R.: Multilevel Algorithms for Mixed Problems. II. Treatment of the Mini-Element, SIAM J. Numer. Anal., vol. 25, no. 2, 1988, pp. 285–293.

11. Wittum, G.: Multi-Grid Methods for Stokes and Navier- Stokes Equations, Transforming Smoothers: Algorithms and Numerical Results, Numer. Math., vol. 54, 1989, pp. 546–563.

12. Wittum, G.: On the Convergence of Multi-Grid Methods with Transforming Smoothers, Theory with Applications to the Navier-Stokes Equations, Numer. Math., vol. 57, 1990, pp. 15–38.

13. Yavneh, I.: Multigrid techniques for Incompressible Flows, Ph.D. Thesis, The Weizmann Institute of Science, Rehovot, 1991.

14. Brezzi, F.; and Fortin, M.: Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, 1991.

15. Girault, V.; and Raviart, P.-A.: Finite Element Methods for Navier-Stokes Equations, Lecture Notes in Mathematics 749, Springer-Verlag, Berlin, 1981.

16. Girault, V.; and Raviart, P.-A.: Finite Element Methods for Navier-Stokes Equations, Springer-Verlag, Berlin, 1986.

17. Gunzburger, M. D.: Finite Element Methods for Viscous Incompressible Flows, Academic Press, Boston, 1989.

18. Adams, R. A.: Sobolev Spaces, Academic Press, New York, 1975.

19. Dautray, R.; and Lions, J. L.: Mathematical Analysis and Numerical Methods for Science and Technology, Vols. 1–5, Springer-Verlag, Berlin, 1988–92.

20. Elman, H. C.: Multigrid and Krylov Subspace Methods for the Discrete Stokes Equations, Seventh Copper Mountain Conference on Multigrid Methods, NASA CP-3339, 1996.

21. Elman, H. C.; and Golub, G. H.: Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems, SIAM J. Numer. Anal., vol. 31, no. 6, 1994, pp. 1645–1661.

22. Nicolaides, R. A.: Existence, Uniqueness and Approximation for Generalized Saddle Point Problems, SIAM J. Numer. Anal., vol. 19, no. 2, 1982, pp. 349–357.

23. Bernardi, C.; Canuto, C.; and Maday, Y.: Generalized Inf-Sup Conditions for Spectral Approximation of the Stokes Problem, SIAM J. Numer. Anal., vol. 25, no. 6, 1988, pp. 1237–1271.

24. Gunzburger, M. D.; and Hou, S. L.: Treating Inhomogeneous Essential Boundary Conditions in Finite Element Methods and the Calculations of Boundary Stresses, SIAM J. Numer. Anal., vol. 29, no. 2, 1992, pp. 390–424.

Page intentionally left blank

MULTIGRID ACCELERATION OF TIME-ACCURATE NAVIER-STOKES CALCULATIONS

N. Duane Melson
NASA Langley Research Center
Hampton, VA 23681-0001

Mark D. Sanetrik
Analytical Services and Materials, Incorporated
Hampton, VA 23681-0001

SUMMARY

A numerical scheme to solve the unsteady Navier-Stokes equations is described. The scheme is fully implicit in time and is unconditionally stable (at least for first- and second-order discretizations of the physical time derivatives). With unconditional stability, the choice of the time step is based on the physical phenomena to be resolved rather than limited by numerical stability. This is especially important for high Reynolds number viscous flows, where the spatial variation of grid cell size can be as much as six orders of magnitude.

A multigrid-multiblock, steady-state, three-dimensional Navier-Stokes solver, TLNS3D, was modified to iteratively invert the equations at each physical time step. The implementation of this procedure in TLNS3D is discussed. The implications of applying several popular turbulence models to unsteady flow are also considered. Numerical results are presented to show the application of the scheme to various two-dimensional turbulent flows. The results of a three-dimensional laminar flow calculation are also given.

INTRODUCTION

Although significant progress has been made in the last twenty years to numerically model many physical situations, most numerical schemes are limited to the prediction of steady flows. This limitation is particularly true in the field of computational fluid dynamics (CFD), where solutions to the Navier-Stokes equations for steady flows are now calculated on a regular basis. (See, for example, references [1–3].) An important factor that has lead to the increased use of Navier-Stokes solvers is the recent success in reducing the computer resources necessary to obtain converged solutions. Perhaps the most promising work has been in the use of multigrid acceleration techniques. Convergence to steady state has been shown in O[log(n)] work, where n represents the number of unknowns to be solved. This reduction in computer requirements has made steady-state solutions affordable to the practicing engineer.

However, many physical phenomena (e.g., separated flows, wake flows, buffet) are intrinsically unsteady. The solution of unsteady problems in CFD has been limited to simplified subsets of the Navier-Stokes equations (panel methods, potential-flow solvers, and some limited use of Euler equation solvers). Unsteady Navier-Stokes calculations have been too expensive for routine use.

The present approach is to apply an iterative procedure for the solution of an implicit equation; thus, the approach is called an *iterative-implicit* method. The concept is not new; in fact, many of the methods developed in the field of linear algebra for inverting large matrices are iterative. Within the field of CFD, similar work is discussed by Jameson [4] for unsteady flows and by Taylor, Ng, and Walters [5] for steady-state flows. The present approach is similar to that of Jameson in that a Runge-Kutta-based multigrid method is used to solve the implicit unsteady flow equations. The Navier-Stokes equations have been treated in the present work, and Jameson's implementation has been modified so that the robustness of the scheme is dramatically increased. Later work by Belov, Martinelli, and Jameson [6] has incorporated the modifications used in the present work as given below and in reference [7].

A summary description of the implementation is given below. (Details of the implementation and analysis of the method are given in a previous paper [7].) A discussion of the use of current 'steady' turbulence models is then given. Numerical results from laminar and turbulent two-dimensional test problems are then presented, as well as the results from a three-dimensional laminar calculation.

GOVERNING EQUATIONS

In the present work, a modified version of the thin-layer Navier-Stokes (TLNS) equations is used to model the flow. The equation set is obtained from the complete Reynolds-averaged Navier-Stokes equations by retaining only the viscous diffusion terms normal to the solid surfaces. For a body-fitted coordinate system (ξ, η, ζ) fixed in time, these equations can be written in the conservation-law form as

$$-\frac{\partial}{\partial T}(J^{-1}U) = \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} - \frac{\partial F_v}{\partial \xi} - \frac{\partial G_v}{\partial \eta} - \frac{\partial H_v}{\partial \zeta}, \tag{1}$$

where U represents the conserved variable vector and F, G, and H represent the convective flux vectors. In the above equation set F_v, G_v, and H_v represent the viscous flux vectors in the three coordinate directions (ξ, η, ζ), and J is the Jacobian of the transformation. These equations represent a more general form of the classical thin-layer equations introduced in reference number [8] because the diffusion terms in all three coordinate directions are included in this form. The Euler equations can easily be recovered from equation (1) by simply dropping the last three terms on the right-hand side. The effects of turbulence are modeled through an eddy-viscosity hypothesis. The Baldwin-Lomax [8], Spalart-Allmaras [9], and Menter shear-stress transport [10] turbulence models are currently implemented to provide turbulence closure.

The temporal derivatives are cast as a fully implicit operator in physical time. For first- or second-order discretizations in time, this produces an unconditionally stable scheme, which allows the time-step size to be chosen based on the temporal resolution needed in the solution rather than limited by the numerical stability requirements. The fully implicit terms are iteratively solved with multigrid acceleration rather than direct inversion, which would be too costly for the nonlinear three-dimensional Navier-Stokes equations.

IMPLEMENTATION OF TIME-DEPENDENT METHOD

Original TLNS3D Method

In the original TLNS3D program, a semidiscrete cell-centered finite-volume algorithm, based on a Runge-Kutta time-stepping scheme [1, 11, 12], is used to obtain the steady-state solutions to the TLNS equations. A linear fourth-difference-based and nonlinear second-difference-based artificial dissipation is added to suppress both the odd-even decoupling and the oscillations in the vicinity of shock waves and stagnation points, respectively. Both the scalar and matrix forms of the artificial dissipation models [13] are incorporated.

In the steady-state implementation, the physical time T is replaced by a pseudo time τ, which gives

$$-\frac{\partial}{\partial \tau}(J^{-1}U) = \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} - \frac{\partial F_v}{\partial \xi} - \frac{\partial G_v}{\partial \eta} - \frac{\partial H_v}{\partial \zeta}. \tag{2}$$

At steady state, the left-hand side of equation (2) disappears, and the right-hand side (the residual) goes to zero, so that any stable scheme may be used to advance the solution in pseudo time.

In the original TLNS3D program, the solution is advanced with a five-stage Runge-Kutta time-stepping scheme. Three evaluations of the artificial dissipation terms (computed at the odd stages) are used to obtain a larger parabolic stability bound, which allows a higher CFL number in the presence of physical viscous diffusion terms. Such a scheme is computationally efficient for solving both the steady Navier-Stokes and the steady Euler equations. The stability range of the numerical scheme is further increased with the use of an implicit residual smoothing technique that employs grid aspect-ratio-dependent coefficients [1, 14, 15].

The solution is advanced in pseudo time with the maximum allowable time step for each cell. The efficiency of the steady numerical scheme is also significantly enhanced through the use of a multigrid acceleration technique as described in reference [1]. The original TLNS3D program was extensively modified to facilitate solution of the flow fields over a wide range of geometric configurations through domain decomposition. This multiblock version of TLNS3D is referred to as TLNS3D-MB. A consequence of this work is the generalization of the boundary conditions of the program to easily accommodate any arbitrary grid topology. A detailed description of this capability is given in reference [16].

In the steady-state version of TLNS3D-MB, the following multistage Runge-Kutta scheme is used to solve (2):

$$
\begin{aligned}
W^{(0)} &= W^{m} \\
&\vdots \\
W^{(k)} &= W^{(0)} + \alpha_k \Delta\tau J^{-1}\left[C^{(k-1)}(W) - D_p^{(\tilde{k})}(W) - D_a^{(\tilde{k})}(W) + F^{(k-1)}(W)\right] \\
&\vdots \\
W^{m+1} &= W^{(K)},
\end{aligned}
\tag{3}
$$

where W is the solution vector for the discrete formulation, m is the counter for the Runge-Kutta iterations, (k) is the k^{th} of K Runge-Kutta stages, α_k is the coefficient for the k^{th} Runge-Kutta stage, C is the convective operator (evaluated at the previous Runge-Kutta stage), D_p and D_a are the physical and artificial dissipation operators (evaluated at a linear combination of previous Runge-Kutta stages), and F is the multigrid forcing function. The above solution procedure can be thought of as placing the equation to be solved (in this case the steady-state Navier-Stokes equations) on the right-hand side of the equation and adding a pseudo-time term on the left-hand side. (See equation (2).) The same type of procedure is used in the time-accurate version of TLNS3D-MB. In this case, however, the unsteady Navier-Stokes equations are placed on the right-hand side:

$$
-\frac{\partial}{\partial\tau}\left(J^{-1}U\right) = \frac{\partial}{\partial t}\left(J^{-1}U\right) + \frac{\partial F}{\partial\xi} + \frac{\partial G}{\partial\eta} + \frac{\partial H}{\partial\zeta} - \frac{\partial F_v}{\partial\xi} - \frac{\partial G_v}{\partial\eta} - \frac{\partial H_v}{\partial\zeta}.
\tag{4}
$$

The physical time derivative is then approximated as a finite difference, and the same type of Runge-Kutta scheme is used to advance the solution in pseudo-time:

$$
\begin{aligned}
W^{(0)} &= W^{m} \\
&\vdots \\
W^{(k)} &= W^{(0)} + \\
\alpha_k \Delta\tau J^{-1}&\left[C^{(k-1)}(W) - D_p^{(\tilde{k})}(W) - D_a^{(\tilde{k})}(W) + F^{(k-1)}(W) - \frac{1}{J^{-1}}\frac{W^{(k)} - W^{n}}{\Delta t}\right] \\
&\vdots \\
W^{m+1} &= W^{(K)},
\end{aligned}
\tag{5}
$$

where n is the physical time step counter. Note that for simplicity the physical time derivative has been written as a first-order derivative; a higher order discretization can be used if more accuracy is desired. Also note that all terms in (5), except for the second term in the physical time derivative, are evaluated at the new physical time level $n + 1$.

Equation (5) cannot be solved directly because the term $W^{(k)}$ appears on both sides of the equation. Solving (5) for $W^{(k)}$ gives

$$\begin{aligned}&(1+\alpha_k\lambda)W^{(k)} = \\ &W^{(0)} + \alpha_k \Delta\tau J^{-1}\left[C^{(k-1)}(W) - D_p^{(\tilde{k})}(W) - D_a^{(\tilde{k})}(W) + F^{(k-1)}(W) + \frac{1}{J^{-1}}\frac{W^n}{\Delta t}\right],\end{aligned} \tag{6}$$

where λ is the ratio of pseudo and physical times $\frac{\Delta\tau}{\Delta t}$. However, (6) also is unacceptable because the right-hand side does not go to zero as the Runge-Kutta iteration converges. The final form for the k^{th} Runge-Kutta stage for the time-dependent version of TLNS3D-MB is obtained by adding and subtracting the term $\alpha_k\lambda W^{(k-1)}$ to the right-hand side of the equation:

$$\begin{aligned}&(1+\alpha_k\lambda)W^{(k)} = W^{(0)} + \alpha_k\lambda W^{(k-1)} + \\ &\alpha_k \Delta\tau J^{-1}\left[C^{(k-1)}(W) - D_p^{(\tilde{k})}(W) - D_a^{(\tilde{k})}(W) + F^{(k-1)}(W) - \frac{1}{J^{-1}}\frac{W^{(k-1)} - W^n}{\Delta t}\right].\end{aligned} \tag{7}$$

For second-order discretization of the physical time derivative, this becomes:

$$\begin{aligned}&\left(1+\frac{3}{2}\alpha_k\lambda\right)W^{(k)} = W^{(0)} + \alpha_k\lambda W^{(k-1)} + \\ &\alpha_k \Delta\tau J^{-1}\left[C^{(k-1)}(W) - D_p^{(\tilde{k})}(W) - D_a^{(\tilde{k})}(W) + F^{(k-1)}(W) - \frac{1}{J^{-1}}\frac{3W^{(k-1)} - 4W^n + W^{n-1}}{2\Delta t}\right].\end{aligned} \tag{8}$$

The Baldwin-Lomax turbulence model is considered a zero-equation turbulence model and is implemented as part of the solution of the Navier-Stokes equations. The one- and two-equation turbulence models are implemented such that their solution is decoupled from the Navier-Stokes equations. They do not contain physical time derivatives and are not treated in a time-accurate manner. From a heuristic standpoint, they can be considered frozen in time. The results presented below indicate that this is an acceptable implementation for the class of problems considered. Subsequent work [17] has indicated that the physical time derivatives should be included in the turbulence model to insure accuracy for a wide range of flows.

NUMERICAL RESULTS

To demonstrate the capability of the present method, the results of several numerical experiments are given. The first case that is examined is the unsteady flow over a two-dimensional circular cylinder with a Reynolds number of 3000 and a free-stream Mach number of 0.2. If the flow about the cylinder is impulsively started, the initial flow is symmetric with zero lift as the wake behind the cylinder begins to grow. As the wake continues to grow, it becomes unstable and begins to shed from alternate sides of the cylinder. This shedding is periodic in nature and is characterized by the Strouhal number. The experimentally obtained value of the Strouhal number for the above conditions is 0.21.

The present scheme was used to calculate the fully developed vortex shedding flow around the cylinder. Two different grids were generated for the calculations; a fine grid with 257 x 129 points around and normal to the cylinder, respectively, and a coarse grid generated by deleting every other point from the fine grid. The fine grid was generated using an algebraic method with simple power law stretching. The normal spacing at the cylinder for the fine grid was 0.0001 times the diameter of the cylinder, and the grid extended to 20 diameters from the center of the cylinder. The coarse grid had a normal spacing of 0.0002 with the same outer boundary. Points were clustered in the wake region for better resolution, as shown for the 'coarse grid in figure 1. Results were obtained for two time step sizes for both the Baldwin-Lomax and Spalart-Allmaras turbulence models. Second-order discretization of physical time was used for all unsteady calculations. The larger, nondimensional time step size of 0.4 gave approximately 50 time steps per period. The smaller time step of 0.2 gave approximately 100 steps per cycle. The predicted Strouhal number for each combination of grid, time step, and turbulence model is presented in table 1. The percent difference from the experimental value is given in parentheses. As would be expected for separated flow, the Spalart-Allmaras turbulence model produced more accurate results for each grid/time step combination. Time histories of the lift coefficient C_l are shown in figures 2 and 3 for the Baldwin-Lomax and Spalart-Allmaras turbulence models. The small effect of the reduction of the time step size indicates that the larger time step (with 50 time steps per cycle) is adequate to predict the Strouhal number. The difference in the results due to the change in grid spacing is much larger than the effect due to changes in the time step size.

Table 1. Predicted Strouhal Number for Circular Cylinder (M_∞ = 0.2, Re_d = 3000)

	Baldwin-Lomax		Spalart-Allmaras	
	coarse grid	fine grid	coarse grid	fine grid
Δt = 0.40	0.197 (6.2%)	0.201 (4.3%)	0.211 (4.8%)	0.207 (1.4%)
Δt = 0.20	0.198 (5.7%)	0.202 (3.8%)	0.219 (4.3%)	0.208 (1.0%)

The second configuration considered was a two-dimensional rectangular cavity in a flat plate. To model a configuration tested experimentally [18], a cavity length of 3.0 inches and height of 0.5 inches were considered. The flat plate extended 10.4 inches upstream of the cavity. This gives a length to height ratio (L/H) of 6. A free-stream Mach number of 0.3 and a Reynolds number of 300,000/inch were used. A transition grit was applied near the plate leading edge to force the boundary layer to transition to a turbulent boundary layer and for these conditions, no tones were generated and the flow was nearly steady.

A nonreflecting boundary condition was applied at the inflow boundary 21.6 inches ahead of the cavity. (See figure 4.) The upper computational boundary was set at 10 inches above the plate where a nonreflecting boundary condition was applied. An extrapolation boundary condition was applied at the outflow boundary 39.1 inches aft of the cavity. An algebraic grid generation technique was used to generate a two-block grid with 49 x 56 points in the cavity and 129 x 49 points above the cavity and flat plate. Power law stretching was used to cluster points near the flat plate and the cavity walls and floor with a spacing of 0.005 inches. A cosine function was used to transition from the clustered grid near the surface to a specified fraction of uniform spacing near the far boundaries. (See figure 5.)

To obtain reasonable starting conditions, TLNS3D-MB was run in steady mode (pseudo-time marching). After a reasonable number of multigrid cycles, the calculation was stopped and then restarted in unsteady mode with second-order physical time discretization. It has been found that this is an effective method for starting unsteady calculations. The lift histories for a laminar calculation and turbulent calculations using the Baldwin-Lomax, Spalart-Allmaras, and Menter models are shown in figure 6. Note that the laminar results exhibit periodic behavior, while the turbulent results appear to approach a steady solution. The turbulent cases were all started from an unsteady laminar solution to try to force oscillations, but all models showed a damping of the oscillations. Detailed examination of the solution shows small oscillations, but the predicted flow is essentially steady. This result is in line with experimental observations of the differences between laminar and turbulent flows in cavities [19–21]. The topology of the flow field in the cavity predicted by the turbulent runs is characterized by a large recirculation region that fills most of the cavity. Small secondary vortices are also present in the lower corners of the cavity. A sample of this is shown in figure 7. The topology of the laminar solution is very different. Multiple nonstationary vortices appear in the cavity and then either die out or are convected out of the cavity. Streaklines at various times are shown in figures 8 and 9.

The computed pressure coefficient along the centerline of the floor of the cavity from the present turbulent calculations is compared with experimental values in figure 10. Once again, the agreement for the Baldwin-Lomax model is not as good as for the one- or two-equation turbulent models for separated flow. None of the models predicts the high pressure at the rear of the cavity as seen in the experimental data. This result may be due to three-dimensional effects in the experiment.

To demonstrate the capability of the present method to calculate three-dimensional flows, a three-dimensional laminar calculation was performed for the same L/H = 6 cavity with a width to height ratio (W/H) of 5. The surface grid and a portion of the outer boundary for this calculation are shown in figure 11. The two-dimensional grid shown previously is the grid from the cavity centerline plane from this three-dimensional grid. The lift and drag (based on integrated pressures) histories of this calculation are shown in figure 12. The flow exhibits the same unsteady properties that the two-dimensional laminar calculation contained, although large three-dimensional effects are apparent, as evidenced by the streaklines for a selected time shown in figure 13. This calculation required approximately 50 CPU hours on a Cray C-90.

CONCLUSIONS

A method to accurately calculate solutions to the unsteady Navier-Stokes equations has been presented. Multigrid acceleration has been successfully employed to accelerate the calculations of the *iterative-implicit* method. Examples for two-dimensional turbulent flow past a circular cylinder and a rectangular cavity, using the Baldwin-Lomax, Spalart-Allmaras, and Menter shear-stress transport models, have been presented to show that a frozen implementation of these ‘steady’ turbulence models can give good results for these unsteady separated flows. The time-dependent scheme has also been demonstrated for a three-dimensional laminar calculation.

REFERENCES

[1] V. N. Vatsa and B. W. Wedan. Development of an Efficient Multigrid Code for 3–D Navier-Stokes Equations and its Application to a Grid-Refinement Study. *Computers and Fluids*, 18(4):391–403, 1990.

[2] F. Ghaffari, J. M. Luckring, J. L. Thomas, and B. L. Bates. Navier-Stokes Solutions About the F/A-18 Forebody-Leading-Edge Extension Configuration. *Journal of Aircraft*, 27:737–748, 1990.

[3] H. L. Atkins. A Multi-Block Multigrid Method for the Solution of the Euler and Navier-Stokes Equations for the Three-Dimensional Flows. AIAA Paper 91–0101, 1991.

[4] A. Jameson. Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings. AIAA Paper 91–1596, 1991.

[5] A. C. Taylor III, W.-F. Ng, and R. W. Walters. Upwind Relaxation Methods for the Navier-Stokes Equations Using Inner Iterations. *Journal of Computational Physics*, 99:68–72, 1992.

[6] A. Belov, L. Martinelli, and A. Jameson. A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations. AIAA Paper 95–0049, 1995.

[7] N. D. Melson, M. D. Sanetrik, and H. L. Atkins. Time-Accurate Navier-Stokes Calculations with Multigrid Acceleration. In NASA CP 3224, (N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds.), pages 423–437, 1993. Presented at the Sixth Copper Mountain Conference on Multigrid Methods.

[8] B. S. Baldwin and H. Lomax. Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. AIAA Paper 78–257, 1978.

[9] P. R. Spalart and S. R. Allmaras. A One-Equation Turbulence Model for Aerodynamic Flows. AIAA Paper 92–0439, 1992.

[10]Florian R. Menter. Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows. AIAA Paper 93–2906, 1993.

[11]A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. AIAA Paper 81–1259, 1981.

[12]A. Jameson and T. J. Baker. Solutions of the Euler Equations for Complex Configurations. AIAA Paper 83–1929, 1983.

[13]E. Turkel and V. N. Vatsa. Effect of Artificial Viscosity of Three Dimensional Flow Solutions. AIAA Paper 90–1444, 1990.

[14]L. Martinelli. *Calculation of Viscous Flows with Multigrid Methods*. PhD thesis, MAE Dept., Princeton University, 1987.

[15]R. C. Swanson and E. Turkel. Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations. AIAA Paper 87–1107, 1987.

[16]V. N. Vatsa, M. D. Sanetrik, and E. B. Parlette. Development of a Flexible and Efficient Multigrid-Based Multiblock Flow Solver. AIAA Paper 93–0677, 1993.

[17]C. L. Rumsey, M. D. Sanetrik, R. T. Biedron, N. D. Melson, and E. B. Parlette. Efficiency and Accuracy of Time-Accurate Turbulent Navier-Stokes Computations. *Computers and Fluids*, 25(2):217–236, 1996.

[18]M. Tracy. Personal communication.

[19]N. M. Komerath, K. K. Ahuja, and F. W. Chambers. Prediction and Measurement of Flows Over Cavities–A Survey. AIAA Paper 87–0166, 1987.

[20]L. W. Shaw. Supersonic Flow Induced Cavity Acoustics. The Shock and Vibration Center, Naval Research Laboratory, Washington, D.C., 1986. Bulletin 56.

[21]H. H. Heller, G. Holmes, and E. E. Covert. Flow-Induced Pressure Oscillations in Shallow Cavities. Technical Report AFFDL-TR-70–104, Air Force Flight Dynamics Laboratory, 1970.

FIGURES

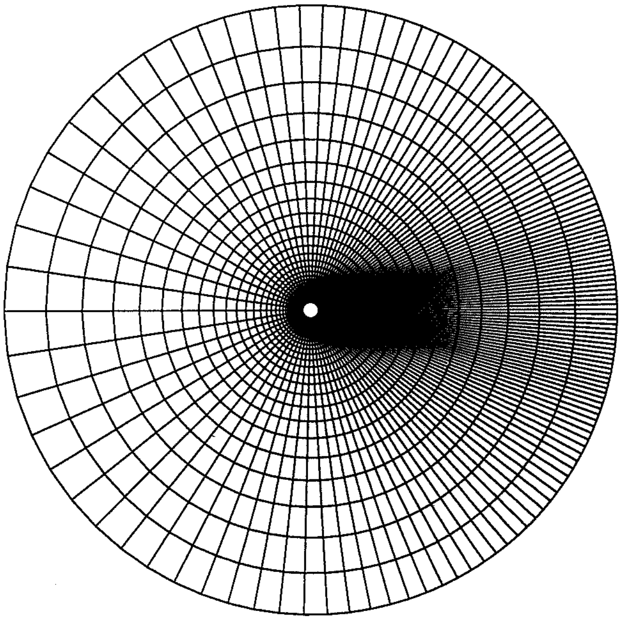


Figure 1. Coarse cylinder grid (129 x 65).

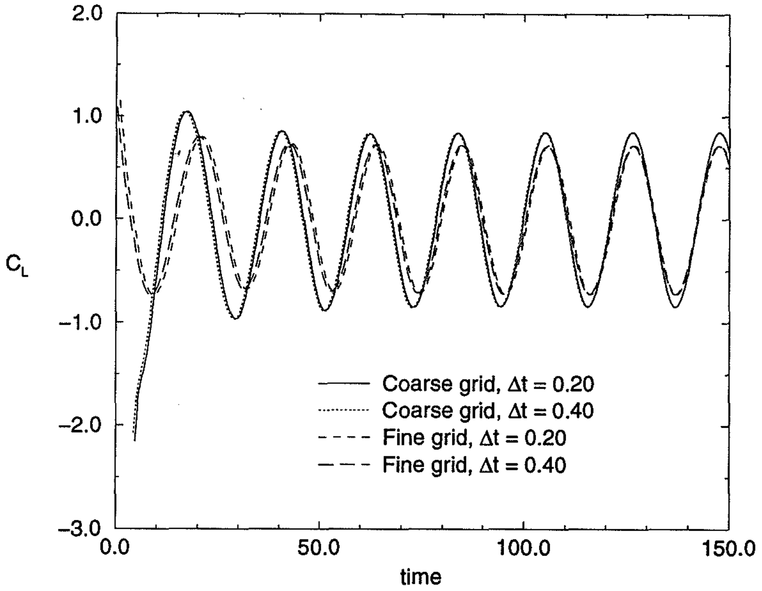


Figure 2. Lift history for circular cylinder with Baldwin-Lomax turbulence model (M_∞=0.2, Re_D=3000).

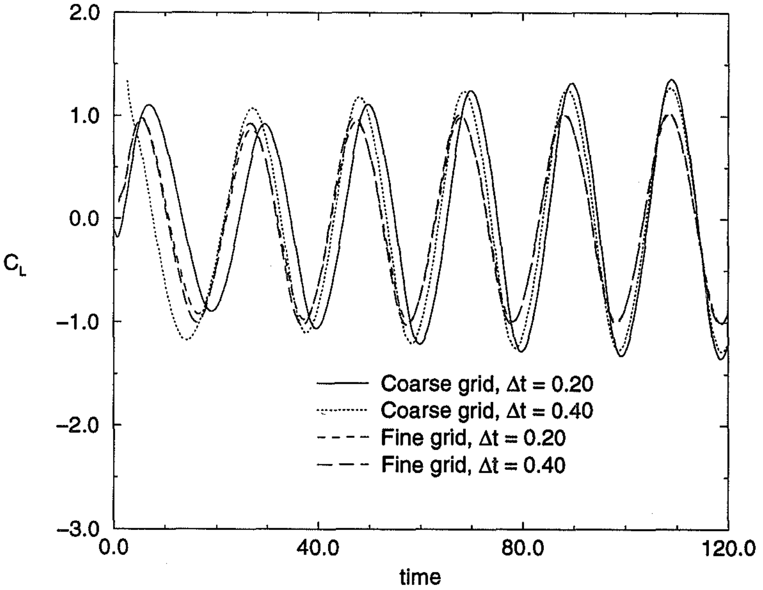


Figure 3. Lift history for circular cylinder with Spalart-Allmaras turbulence model (M_∞=0.2, Re_D=3000).

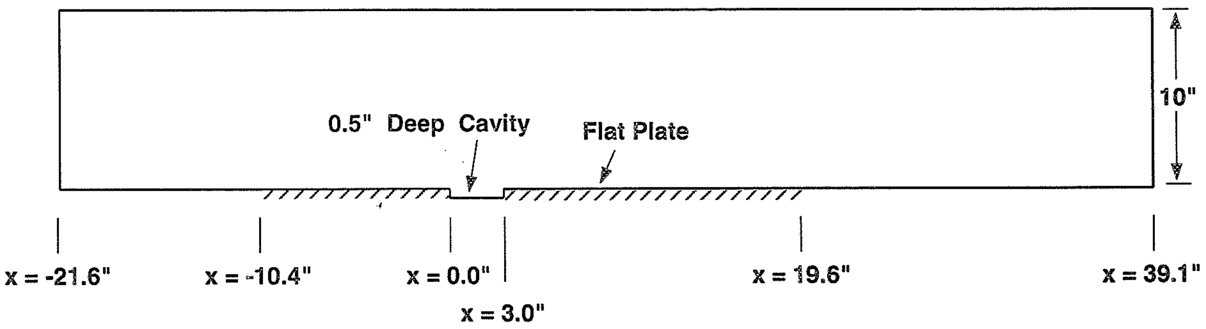


Figure 4. Schematic of two-dimensional rectangular cavity computational domain.

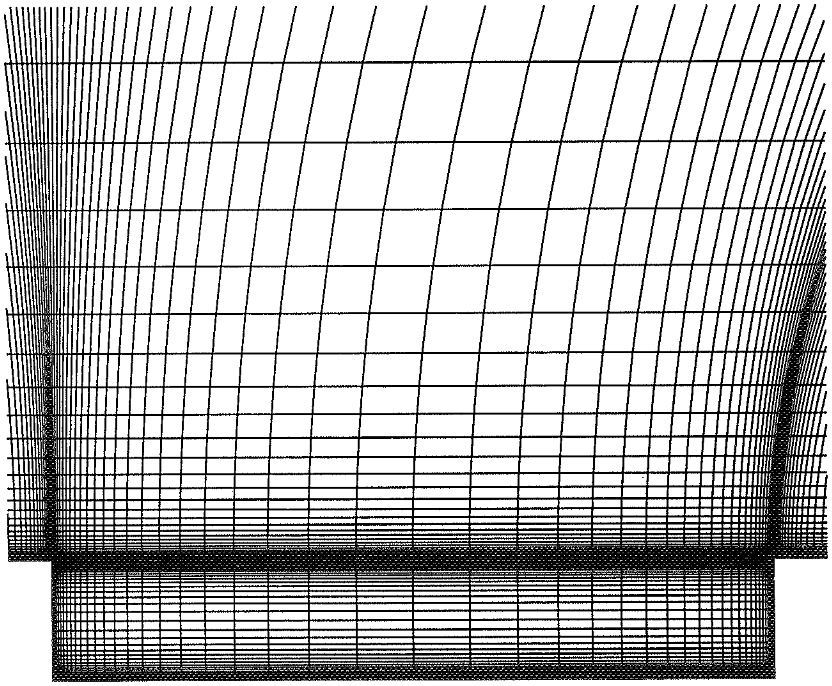


Figure 5. Grid for two-dimensional rectangular cavity calculations (L/H=6).

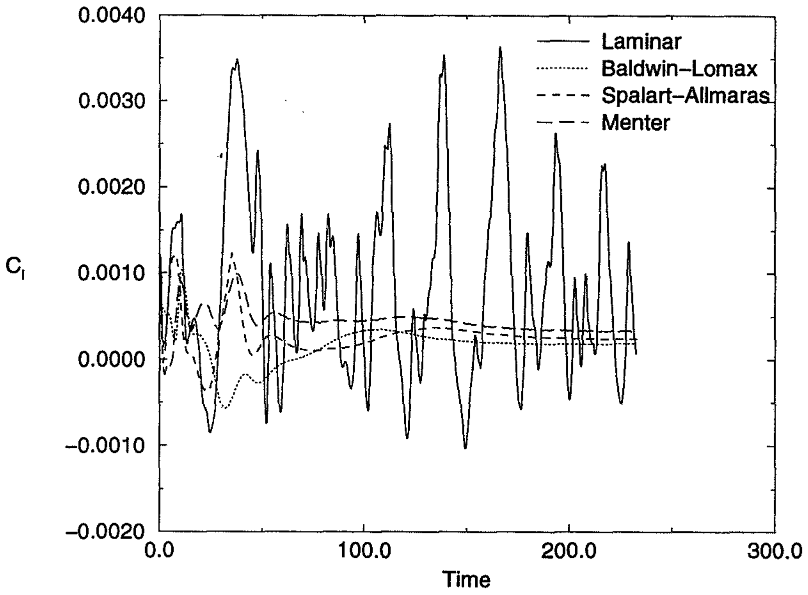


Figure 6. Lift history for two-dimensional rectangular cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

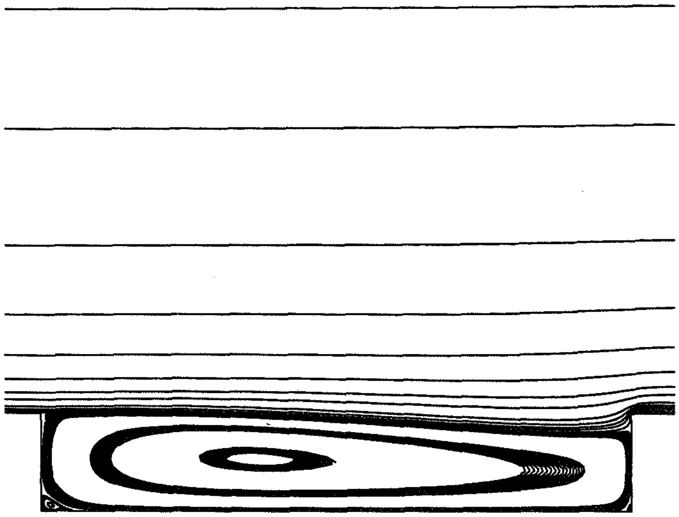


Figure 7. Sample streaklines for turbulent (Spalart-Allmaras) calculation of two-dimensional cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

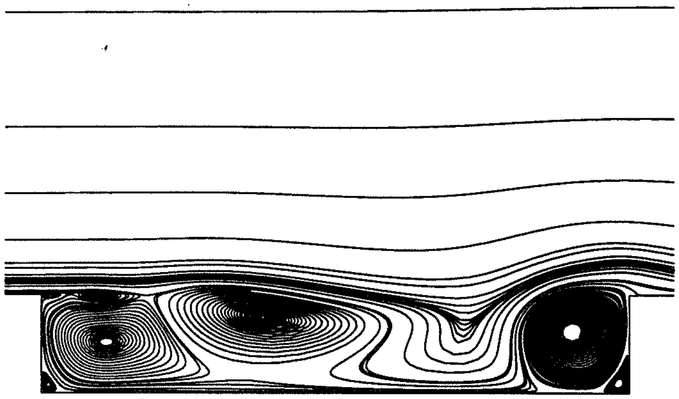


Figure 8. Sample streaklines at T=109.5 for laminar calculation of two-dimensional cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

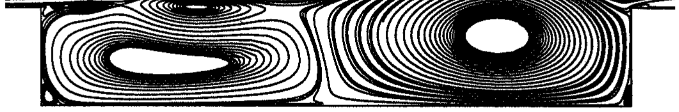


Figure 9. Sample streaklines at T=120.75 for laminar calculation of two-dimensional cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

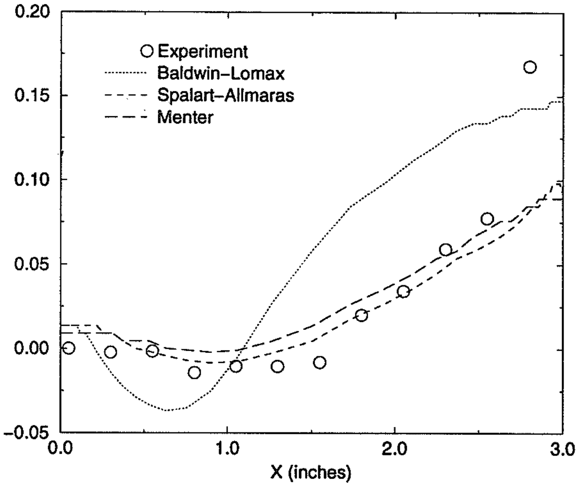


Figure 10. Pressure coefficient along cavity floor for two-dimensional rectangular cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

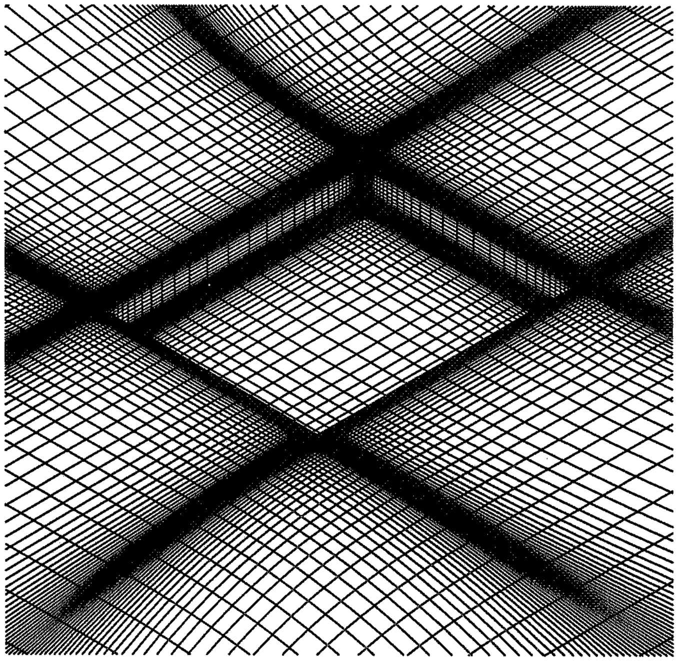


Figure 11. Surface grid for three-dimensional rectangular cavity calculations (L/H=6, W/H=5).

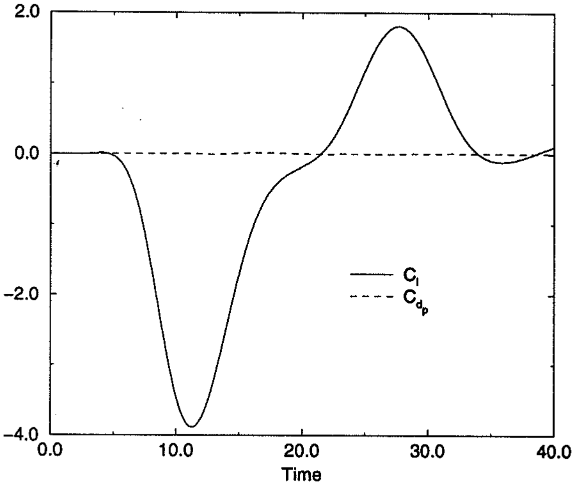


Figure 12. Lift and drag coefficient histories for three-dimensional rectangular cavity (L/H=6, M_∞=0.4, Re=300,000/inch).

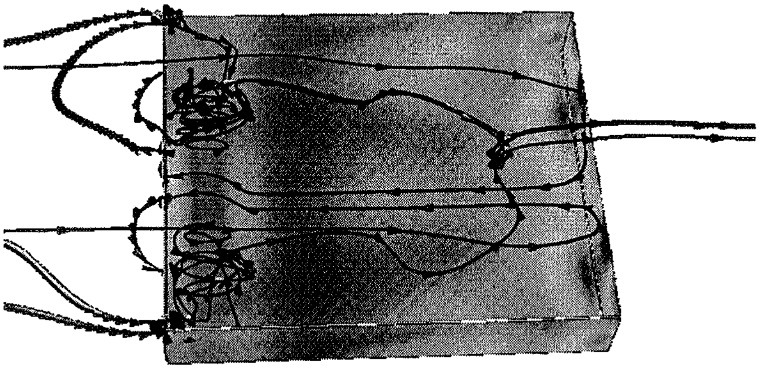


Figure 13. Sample streaklines for laminar calculation of three-dimensional cavity (L/H=6, W/H=5, M_∞=0.4, Re=300,000/inch).

Page intentionally left blank

MULTIGRID METHODS FOR FULLY IMPLICIT OIL RESERVOIR SIMULATION

J. Molenaar
TWI, Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

INTRODUCTION

In this paper we consider the simultaneous flow of oil and water in reservoir rock. This displacement process is modeled by two basic equations (see, e.g., [1]): the material balance or continuity equations and the equation of motion (Darcy's law). For the numerical solution of this system of nonlinear partial differential equations there are two approaches: the fully implicit or simultaneous solution method and the sequential solution method.

In the sequential solution method the system of partial differential equations is manipulated to give an elliptic pressure equation and a hyperbolic (or parabolic) saturation equation. In the IMPES approach the pressure equation is first solved, using values for the saturation from the previous time level. Next the saturations are updated by some explicit time stepping method; this implies that the method is only conditionally stable. For the numerical solution of the linear, elliptic pressure equation multigrid methods have become an accepted technique. (See, e.g., [2],[3],[4].)

On the other hand, the fully implicit method is unconditionally stable, but it has the disadvantage that in every time step a large system of nonlinear algebraic equations has to be solved. The most time-consuming part of any fully implicit reservoir simulator is the solution of this large system of equations. Usually this is done by Newton's method. The resulting systems of linear equations are then either solved by a direct method or by some conjugate gradient type method.

In this paper we consider the possibility of applying multigrid methods for the iterative solution of the systems of nonlinear equations. There are two ways of using multigrid for this job: either we use a nonlinear multigrid method or we use a linear multigrid method to deal with the linear systems that arise in Newton's method. So far only a few authors have reported on the use of multigrid methods for fully implicit simulations. In [5] a two-level FAS algorithm is presented for the black-oil equations, and linear multigrid for two-phase flow problems with strong heterogeneities and anisotropies is studied in [6]. Here we consider both possibilities. Moreover we present a novel way for constructing the coarse grid correction operator in linear multigrid algorithms. This approach has the advantage in that it preserves the sparsity pattern of the fine grid matrix and it can be extended to systems of equations in a straightforward manner. We compare the linear and nonlinear multigrid algorithms by means of a numerical experiment.

EQUATIONS

In the absence of gravity forces the volumetric flow rate of water and oil in a porous medium is given by the generalized Darcy's law

$$q_\alpha = -\lambda_\alpha \nabla P_\alpha, \quad \alpha = w, o, \tag{1}$$

where q_α, λ_α, and P_α are the Darcy velocity, the mobility, and the pressure of phase α, respectively. The saturation of phase α is denoted by S_α, so

$$S_w + S_o = 1. \tag{2}$$

The phase mobilities λ_α are defined by

$$\lambda_\alpha = k\frac{k_\alpha}{\mu_\alpha}, \quad \alpha = w, o, \tag{3}$$

where k is the rock permeability, $k_\alpha(S_\alpha)$ is the phase relative permeability, and μ_α is the phase viscosity. In addition to these momentum equations we have mass conservation laws for both phases:

$$\phi\frac{\partial S_\alpha}{\partial t} + \nabla \cdot q_\alpha + Q_\alpha = 0, \quad \alpha = w, o, \tag{4}$$

where ϕ is the porosity of the rock and Q_α is the production rate of phase α. The phase pressures P_α are related through the capillary pressure P_c:

$$P_c(S_w) = P_o - P_w. \tag{5}$$

The equations (1)-(5) are the partial differential equations that make up the incompressible two-phase flow model. In the sequel we use S_w and P_o as the independent variables and drop the subscripts.

We still have to specify the boundary conditions. Usually the flow across well boundaries is modeled by point sources and sinks, and no flow boundary conditions are imposed at the boundary of the reservoir. This has the effect of shifting all complications to a proper modeling of the injection and production wells.

DISCRETIZATION

In this section we describe the fully implicit discretization of the multiphase flow equations. For ease of notation we assume a uniform porosity ϕ and rock permeability k. Moreover we only consider the two-dimensional case with a uniform Cartesian grid. The equations are discretized in space by a finite volume scheme (cell-centered finite-differences or box scheme). For the time integration the backward Euler method is used. This leads to the system of equations

$$\begin{aligned} &+\frac{\phi}{\Delta t}\left(S_{i,j}^{n+1} - S_{i,j}^{n}\right) + (Q_w)_{i,j}^{n+1} + \\ &\quad \frac{1}{h}\left((q_w)_{i+\frac{1}{2},j}^{n+1} - (q_w)_{i-\frac{1}{2},j}^{n+1} + (q_w)_{i,j+\frac{1}{2}}^{n+1} - (q_w)_{i,j-\frac{1}{2}}^{n+1}\right) = 0, \end{aligned} \tag{6}$$

$$\begin{aligned} &-\frac{\phi}{\Delta t}\left(S_{i,j}^{n+1} - S_{i,j}^{n}\right) + (Q_o)_{i,j}^{n+1} + \\ &\quad \frac{1}{h}\left((q_o)_{i+\frac{1}{2},j}^{n+1} - (q_o)_{i-\frac{1}{2},j}^{n+1} + (q_o)_{i,j+\frac{1}{2}}^{n+1} - (q_o)_{i,j-\frac{1}{2}}^{n+1}\right) = 0. \end{aligned} \tag{7}$$

In the above, h denotes the mesh width; the subscripts i, j, the discretization cell; and the superscript n, the time level. The fluxes at the edges between cells are approximated with upstream weighted mobilities. For example, the fluxes $(q_\alpha)_{i+\frac{1}{2},j}^{n+1}$ at the edge between the cells i, j and $i, j+1$ are approximated by

$$(q_w)_{i+\frac{1}{2},j}^{n+1} = -(\lambda_w)_{i+\frac{1}{2},j}^{n+1}\frac{P_{i+1,j}^{n+1} - (P_c)_{i+1,j}^{n+1} - P_{i,j}^{n+1} + (P_c)_{i,j}^{n+1}}{h}, \tag{8}$$

$$(q_o)_{i+\frac{1}{2},j}^{n+1} = -(\lambda_o)_{i+\frac{1}{2},j}^{n+1}\frac{P_{i+1,j}^{n+1} - P_{i,j}^{n+1}}{h}, \tag{9}$$

with

$$(\lambda_\alpha)_{i+\frac{1}{2},j}^{n+1} = \begin{cases} k_{i+\frac{1}{2},j}\frac{k_\alpha(S_i^{n+1})}{\mu_\alpha}, & \text{if } (P_\alpha)_{i+1,j}^{n+1} - (P_\alpha)_{i,j}^{n+1} \leq 0, \\ k_{i+\frac{1}{2},j}\frac{k_\alpha(S_{i+1}^{n+1})}{\mu_\alpha}, & \text{if } (P_\alpha)_{i+1,j}^{n+1} - (P_\alpha)_{i,j}^{n+1} > 0. \end{cases} \tag{10}$$

In the case of nonuniform rock permeability k, the permeability $k_{i+\frac{1}{2},j}$ at the cell edge is the harmonic average of the values in the adjacent cells.

In each time step we have to solve the large system of nonlinear equations (6)-(10). We consider cell-centered multigrid methods for the iterative solution of these systems. In cell-centered multigrid methods the coarser grids G^{2h}, G^{4h}, $\cdots$ are constructed by successively doubling the mesh width of the fine grid G^h. Hence, each coarse grid cell is the union of four fine grid cells. In this paper we focus on the coarse grid correction. Suppose that on the fine grid G^h we have the system of equations

$$\mathcal{N}^h(u^h) = f^h, \tag{11}$$

where $\mathcal{N}^h$ is a possibly nonlinear operator. The coarse grid corrections that we consider are of the form

$$\mathcal{N}^{2h}(\tilde{u}^{2h}) = \mathcal{N}^{2h}(u^{2h}) + \overline{R}^h_{2h}(f^h - \mathcal{N}^h(u^h)), \tag{12}$$

$$\tilde{u}^h = u^h + \tilde{P}^{2h}_h(\tilde{u}^{2h} - u^{2h}), \tag{13}$$

where $\overline{R}^h_{2h}$ denotes the restriction that is the adjoint of the interpolation by a piecewise constant function. In the cell centered multigrid method this is natural: the residual (the total excess of accumulation and net flow) in a coarse grid cell is the sum of the residuals in the corresponding four fine grid cells. The prolongation $\tilde{P}^{2h}_h$ is the piecewise bilinear interpolation. This combination of prolongation and restriction is formally sufficiently accurate to deal with second order partial differential equations.

We will now develop two multigrid methods for (6)-(10). In the nonlinear multigrid method (the FAS algorithm [7]) we deal with this nonlinear system of equations directly, so $\mathcal{N}^h$ is a nonlinear operator. On the other hand, in the linear multigrid method $\mathcal{N}^h$ is the Jacobian matrix of the system of nonlinear equations. We present a novel way to construct the coarse grid correction operator for the linear multigrid algorithm.

Nonlinear Multigrid

The nonlinear multigrid method that we use is the FAS algorithm. To obtain the coarse grid operator $\mathcal{N}^{2h}$ the problem is discretized on the coarse grid (i.e., a grid with mesh size $2h$). There are only homogeneous boundary conditions; therefore, the treatment of the boundary conditions on the coarse grids is trivial. If there is a well in a grid cell on the fine grid, then it is also present in all father cells on coarser grids. Because the problem is nonlinear, the properties of the coarse grid operators are determined by the choice of u^{2h}. Here we take $u^{2h} = R^h_{2h}u^h$, where R^h_{2h} is the interpolation by piecewise constants.

We use a collective point Gauss-Seidel-Newton method as the smoother in this multigrid algorithm. This means that all cells are visited in some predetermined order, and equations (6) and (7) are solved simultaneously for the variables related to that cell. This system of two nonlinear equations is solved by Newton's method.

Linear Multigrid

We can also use multigrid to solve the linear systems of equations that occur when applying Newton's method on the fine grid G^h. Let us again consider the construction of the coarse grid linear operator that is used in the coarse grid correction. Basically there are two ways to define this coarse grid operator. Given prolongation and restriction operators we can define the coarse grid operator as the Galerkin approximation to the fine grid operator; this is done in [6]. This approach is straightforward but it has a disadvantage in that for simple linear elliptic equations the coarse grid matrix may loose the M-matrix property. Moreover, the stencils of the coarse grid operators are often denser than the corresponding fine grid stencil (cf. [8]). The alternative approach is to discretize the problem on the coarse grid as in the nonlinear multigrid algorithm and to use the Jacobian of the nonlinear coarse grid operator as the coarse grid operator for the linear multigrid algorithm. An advantage to this approach is that all nice properties of the fine grid operator are immediately carried over to the operators on the coarser grids. We now try to combine these approaches; the coarse grid operator is defined by means of a Galerkin-like construction that is based on the coarse grid discretization approach.

To explain this construction we consider a simple one-dimensional, second-order scalar conservation law

$$\frac{\mathrm{d}q}{\mathrm{d}x} = f, \tag{14}$$

where q is some function of the solution u and $\frac{\mathrm{d}u}{\mathrm{d}x}$. A simple finite volume discretization on the fine grid G^h with uniform mesh width h leads to a system of equations of the form

$$q^h_{i+\frac{1}{2}} - q^h_{i-\frac{1}{2}} = hf^h_i, \tag{15}$$

with

$$q^h_{i+\frac{1}{2}} = q^h_{i+\frac{1}{2}}(u^h_i, u^h_{i+1}, h). \tag{16}$$

Suppose that we use Newton's method on the grid G^h. In a single iteration step we then solve the following problem: find Δu^h_i such that

$$hf^h_i - (q^h_{i+\frac{1}{2}} - q^h_{i-\frac{1}{2}}) = \Delta q^h_{i+\frac{1}{2}} - \Delta q^h_{i-\frac{1}{2}}, \tag{17}$$

with

$$\Delta q^h_{i+\frac{1}{2}} = \frac{\partial q^h_{i+\frac{1}{2}}}{\partial u^h_i}\Delta u^h_i + \frac{\partial q^h_{i+\frac{1}{2}}}{\partial u^h_{i+1}}\Delta u^h_{i+1}. \tag{18}$$

This can be written in matrix form:

$$J^h \Delta u^h = f^h. \tag{19}$$

For example, let us consider the linear convection-diffusion equation

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(u + \epsilon\frac{\mathrm{d}u}{\mathrm{d}x}\right) = 0, \tag{20}$$

with boundary conditions $u(0) = 0$ and $u(1) = 1$. A forward discretization for the convective term yields

$$q^h_{i+\frac{1}{2}}(u^h_i, u^h_{i+1}, h) = u^h_{i+1} + \epsilon\frac{u^h_{i+1} - u^h_i}{h}. \tag{21}$$

If we use discretization on the coarse grid G^{2h} to define the coarse grid operator, its stencil is given by

$$\left(\frac{\epsilon}{2h},\ -1 - \frac{\epsilon}{h},\ 1 + \frac{\epsilon}{2h}\right). \tag{22}$$

Interpolation by piecewise constants, which is the natural choice for prolongation and restriction in multigrid algorithms for finite volume schemes, is of course insufficiently accurate for this second order problem. However, if we construct the coarse grid operator as the Galerkin approximation using these natural transfer operators, we obtain the coarse grid stencil

$$\left(\frac{\epsilon}{h},\ -1 - \frac{2\epsilon}{h},\ 1 + \frac{\epsilon}{h}\right). \tag{23}$$

Clearly the treatment of the second order diffusion term is different for the finite volume discretization approach (22) and the Galerkin approximation (23).

We compare the efficiency of these two methods by means of a simple numerical experiment. We take the convection-diffusion equation (21) with $\epsilon = 0.01$. In both cases we use a restriction that is the transpose of piecewise constant interpolation and a prolongation by a piecewise linear

h	$\frac{h}{2\epsilon}$	Galerkin		FVD	
		$\nu = 1$	$\nu = 2$	$\nu = 1$	$\nu = 2$
1/8	6.25	0.60	0.38	0.53	0.36
1/16	3.12	0.58	0.42	0.54	0.37
1/32	1.56	0.55	0.44	0.50	0.35
1/64	0.78	0.54	0.45	0.47	0.31
1/128	0.39	0.54	0.47	0.45	0.32
1/256	0.20	0.53	0.47	0.42	0.30
1/512	0.10	0.53	0.47	0.43	0.28

Table 1: Two-level convergence rates for the linear convection-diffusion equation with two different coarse grid operators: the Galerkin approximation and Finite Volume Discretization.

function. For smoothing we apply damped Jacobi relaxation with a damping factor of 2/3. We do not use a Gauss-Seidel smoother because it is an exact solver for the pure convection equation. Therefore, it is not suitable for comparing the merits of the different coarse grid correction operators in the convection dominated case. Table 1 shows the observed two-level convergence rates for both algorithms with one ($\nu = 1$) and two ($\nu = 2$) smoothing steps. If the mesh Peclet number $h/2\epsilon$ is greater than 1 (convection dominates), then the convergence rates are comparable for both algorithms. Applying two smoothing steps improves the convergence rates, so low frequency error components are indeed reduced efficiently in the coarse grid correction. However, when diffusion dominates, the two-grid algorithm with the Galerkin approximation performs worse than the coarse grid discretization approach. Applying two smoothing sweeps hardly improves the convergence rate of the algorithm with the Galerkin approximation. As the grid interpolation operators used in its construction are too inaccurate, the coarse grid correction is incorrect.

Comparing the coarse grid stencils (22) and (23) suggests another approach for the construction of the coarse grid matrix (cf. [8]). Let us assume that we can split the derivatives $\frac{\partial q}{\partial u}$ in terms with different order behavior with respect to the mesh size h:

$$\frac{\partial q^h_{i+\frac{1}{2}}(u^h_i, u^h_{i+1}, h)}{\partial u^h_i} = j_{q_{i+\frac{1}{2}}, u_i}(u^h_i, u^h_{i+1}, h) = \sum_{p=0,1} j^p_{q_{i+\frac{1}{2}}, u_i}(u^h_i, u^h_{i+1}, h), \tag{24}$$

$$\frac{\partial q^h_{i+\frac{1}{2}}(u^h_i, u^h_{i+1}, h)}{\partial u^h_{i+1}} = j_{q_{i+\frac{1}{2}}, u_{i+1}}(u^h_i, u^h_{i+1}, h) = \sum_{p=0,1} j^p_{q_{i+\frac{1}{2}}, u_{i+1}}(u^h_i, u^h_{i+1}, h), \tag{25}$$

with

$$j^p_{q_{i+\frac{1}{2}}, u_i}(u - h\Delta u, u + h\Delta u, h) = \mathcal{O}(h^{-p}) \quad \text{for } h \to 0. \tag{26}$$

For the forward discretization (20) of the linear convection-diffusion equation this leads to a splitting in convective and diffusive terms:

$$j^0_{q_{i+\frac{1}{2}}, u_i} = 0, \quad j^1_{q_{i+\frac{1}{2}}, u_i} = -\frac{\epsilon}{h}, \quad j^0_{q_{i+\frac{1}{2}}, u_{i+1}} = 1, \quad j^1_{q_{i+\frac{1}{2}}, u_{i+1}} = +\frac{\epsilon}{h}. \tag{27}$$

Let the matrix $J^{h,p}$ consist of the elements $j^p_{q_{i+\frac{1}{2}}, u_i}$, so

$$J^h = \sum_{p=0,1} J^{h,p}. \tag{28}$$

We define the coarse grid operator now as follows:

$$J^{2h} = \overline{R}^h_{2h} \left(\sum_{p=0,1} 2^{-p} J^{h,p} \right) P^{2h}_h, \tag{29}$$

where P_h^{2h} and $\overline{R}_{2h}^h$ are the interpolation by piecewise constants. For the example of the linear convection-diffusion equation this yields exactly the same coarse grid operator as the one obtained by discretization on the coarse grid (cf. (22)).

We use this approach for defining the coarse grid operator also in the case of a system of conservation laws. For our two-phase flow model the fluxes are given by (8), (9), and (10). With obvious abuse of notation we define the splitting as follows:

$$j^0_{\alpha,P_{i,j}} = 0, \quad \alpha = w, o, \tag{30}$$

$$j^1_{\alpha,P_{i,j}} = +(\lambda_\alpha)_{i+\frac{1}{2},j}\frac{1}{h}, \quad \alpha = w, o, \tag{31}$$

$$j^0_{w,S_{i,j}} = -\frac{\partial(\lambda_w)_{i+\frac{1}{2},j}}{\partial S_{i,j}} \frac{P_{i+1,j} - (P_c)_{i+1,j} - P_{i,j} + (P_c)_{i,j}}{h}, \tag{32}$$

$$j^1_{w,S_{i,j}} = -(\lambda_w)_{i+\frac{1}{2},j}\frac{1}{h}\frac{\mathrm{d}P_c}{\mathrm{d}S_{i,j}}, \tag{33}$$

$$j^0_{o,S_{i,j}} = -\frac{\partial(\lambda_o)_{i+\frac{1}{2},j}}{\partial S_{i,j}} \frac{P_{i+1,j} - P_{i,j}}{h}, \tag{34}$$

$$j^1_{o,S_{i,j}} = 0. \tag{35}$$

The accumulation terms are of course treated as zero order terms.

We notice that the implementation of (29) is simple due to the fact that we are using piecewise constant grid interpolation operators. The entries of the fine grid matrix consist of terms related to either cells (the accumulation terms) or to edges (the flux terms). The coarse grid matrices have the same structure, where the coarse grid cells consist of four fine grid cells; the coarse grid edges consist of two fine grid edges. Because we are using piecewise constant interpolation operators, (29) implies that we can simply add the terms related to cells on the finest grid to the corresponding terms in parent cells on all coarser grids. Next we calculate the flux terms j^p_{α,S_i} and j^p_{α,P_i}. Each of these terms can be associated with a unique edge between two cells. As we are using piecewise constant interpolation operators and as the terms j^p_{α,S_i} and j^p_{α,P_i} appear with opposite sign in the linearized discrete equations for the two cells (cf. (6) and (7)), it follows that these terms do not contribute to the coarse grid matrix if the fine grid edge is not part of a coarse grid edge. However if the fine grid edge is part of a coarse grid edge, we add that coefficient, multiplied by the appropriate scaling factor, to the coefficient at the parent edge. This is done recursively until we end at the coarsest grid. The splitting in terms related to cells and edges thus yields a straightforward implementation of (29).

NUMERICAL RESULTS

In this section we show some results for the numerical simulation of the flooding of a typical laboratory scale model. This problem is taken essentially from [9]. The model consists of a thin sand pack simulating a quadrant of an infinitely repeating five-spot. Some properties of the model are shown in Table 2. The model is placed horizontally, so the gravity effect can be neglected. Initially there is a uniform saturation S_i in the model. Water is then injected into one corner of the pack at a constant rate q_i, and oil and water are produced at the opposite corner. Several cases are considered with widely varying oil-water viscosity ratios $M = \mu_o/\mu_w$. (See Table 3.) For these data the flow is convection dominated, so steep gradients develop in the water saturation S_w. Because the transition regions cannot be resolved on the coarser grids, this is an interesting test problem for the multigrid algorithms. The functions $k_\alpha(S)$ and $P_c(S)$ are smooth functions and good approximations to the data given in [9]:

$$k_w(S) = \left(\frac{S - 0.2}{0.8}\right)^2, \tag{36}$$

$$k_o(S) = 0.67 \left(\frac{0.9 - S}{0.7}\right)^{1.2}, \tag{37}$$

$$P_c(S) = \left(\frac{S - 0.9}{0.9}\right)^2 62.3 \times 10^3 \text{ [dyne/cm}^2\text{]}. \tag{38}$$

For the discretization of this problem we use several grids. The coarsest grid in all calculations is a 5 × 5 grid, and the fine grid contains 80 × 80 grid points, so the total number of unknowns for the fine grid is 12800. The calculation is stopped when three times the total pore volume has been injected.

In all time steps the discrete problem is solved with a tolerance $\tau < 1 \times 10^{-3}$, where τ denotes the ℓ_2-norm of the residual scaled by the inflow $q_i \Delta t$ in that time step. The total oil balance error ([initial − final oil in place]/cumulative oil production) is always less than 2×10^{-4}. The time steps Δt^n are selected in order to have changes in the saturation of approximately 0.05:

$$\Delta t^{n+1} = \frac{0.05}{||S^n - S^{n-1}||_\infty} \Delta t^n. \tag{39}$$

The ratio $\Delta t^{n+1}/\Delta t^n$ is bounded between 0.5 and 2.0.

In Figure 1 the numerical approximation of the water saturation after injection of 0.25 times the total pore volume is plotted for test problems 1 and 2. In test problem 1 there is a favorable mobility ratio M, and the water displaces the oil in a piston-like manner. However, in test problem 2 we have an unfavorable viscosity ratio. The water saturation at the shock front is now lower than in the previous case, and the water breakthrough occurs earlier. This is in agreement with the classical one-dimensional Buckley-Leverett theory. Figure 2 shows the volume of produced oil versus the volume of injected water expressed in pore volumes. These results are obtained on the 80 × 80 grid. These production curves are (of course) in good agreement with the results presented in [9]. As expected from the Buckley-Leverett theory a large mobility ratio M leads to an inefficient oil recovery process.

For our purposes, the convergence speeds of the two multigrid algorithms that we are considering are more interesting. To estimate the convergence speed of the nonlinear multigrid algorithm, we use the average residual factor ρ_{NMG}. Here we take the ℓ_2-norm of the residual of the *nonlinear* discrete equations ((6) and (7)). The convergence speed of the linear multigrid algorithm is estimated by the average residual factor ρ_{LMG}, which uses the ℓ_2-norm of the residual of the *linear* equations in Newton's method. In all runs we used F-cycles with a single smoothing step for pre- and post-smoothing. Because the flow is basically from the injection corner toward the production corner, a single Gauss-Seidel sweep suffices. In more complicated situations a four direction Gauss-Seidel method has to be used.

In Table 4 we show the estimated convergence speeds ρ on different fine grids for the different test cases. In all cases both multigrid algorithms perform satisfactorily; we observe a fast, grid-independent convergence behavior. The average residual reduction factor ρ is always less than 0.15. In the nonlinear multigrid algorithm we find that typically three or four F-cycles are needed to satisfy the stopping criterion. In the linear multigrid algorithm typically two Newton steps are needed for convergence; altogether, typically four F-cycles per time step are needed. In Table 5 the average execution times on a HP-735 work-station are shown. Although our code is far from optimal, two tentative conclusions can be drawn from it. First, both algorithms show optimal complexity; the time needed per time step and per grid point is independent of the number of grid points. Second, the linear multigrid algorithm is more efficient than the nonlinear one. This is due to the fact that in the nonlinear algorithm functions like $k_\alpha(S)$ and $P_c(S)$ (and their derivatives) have to be calculated much more often.

SUMMARY

We have presented two multigrid algorithms for the fully implicit simulation of incompressible, immiscible two-phase flow in a porous medium. The nonlinear multigrid algorithm is a standard

Side of square	40.64 cm
Thickness	1.27 cm
Porosity	0.375
Permeability	$10.96 \times 10^{-8}\text{cm}^2$

Table 2: Constant data for sand pack model.

	μ_o	μ_w	q_i	S_i
	[cp]	[cp]	[cm/min]	
Test 1	1.37	16.46	10	0.125
Test 2	9.28	1.15	6	0.087
Test 3	162.1	1.15	7	0.087
Test 4	945	1.15	1.5	0.087

Table 3: Data for test problems.

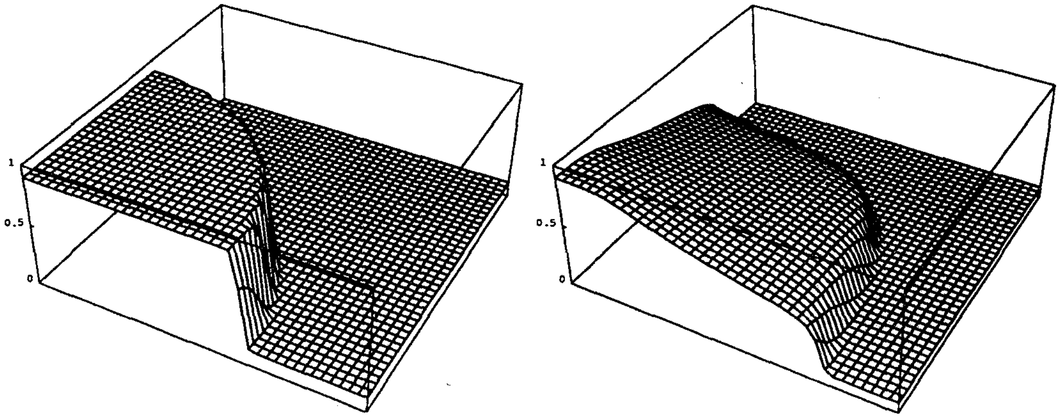


Figure 1: Water saturation for Test 1 (left) and Test 2 (right) after injection of 0.25 pore volume.

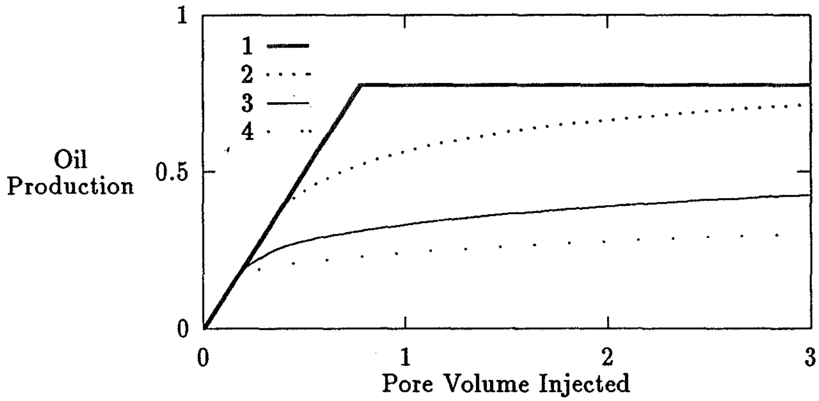


Figure 2: Oil production curves for the different test problems.

grid	LINMLTG				NLMLTG			
	1	2	3	4	1	2	3	4
10 × 10	0.09	0.14	0.17	0.17	0.11	0.14	0.17	0.13
20 × 20	0.10	0.13	0.16	0.16	0.12	0.13	0.16	0.14
40 × 40	0.10	0.12	0.15	0.15	0.12	0.12	0.15	0.14
80 × 80	0.10	0.11	0.14	0.14	0.10	0.11	0.14	0.13

Table 4: Convergence rates for different test cases.

grid	LINMLTG	NLMLTG
10 × 10	0.38	1.53
20 × 20	0.39	1.89
40 × 40	0.50	2.09
80 × 80	0.53	2.09

Table 5: Typical execution times [msec] per time step per grid point.

FAS algorithm. The linear multigrid algorithm that is used to solve linear systems in Newton's method employs a nonstandard construction for the coarse grid matrix. Both algorithms perform satisfactorily for a simple 2D test problem. The linear multigrid algorithm appears to be more efficient with respect to the execution time needed.

REFERENCES

[1] Aziz, K. and Settari, A., *Petroleum Reservoir Simulation*, Elsevier Applied Science Publishers, 1979.

[2] Dendy Jr., J., Two Multigrid Methods For Three-dimensional Problems With Discontinuous and Anisotropic Coefficients, *SIAM J. Sci. Statist. Comput.*, 8:673–685, 1987.

[3] Schmidt, G. and Jacobs, F., Adaptive Local Grid Refinement and Multi-grid in Numerical Reservoir Simulation, *J. Comput. Phys.*, 77:140–165, 1988.

[4] Scott, T., Multi-Grid Methods for Oil Reservoir Simulation in Two and Three Dimensions, *J. Comput. Phys.*, 59:290–307, 1985.

[5] Collins, D. and Mourits, F., Multigrid Methods Applied to Near-Well Modelling in Reservoir Simulation, in *Proceedings ECMOR III*, Christie, M. A. et al., editors, pp. 359–371, Delft University Press, 1992.

[6] Brakhagen, F. and Fogwell, T., Multigrid for the Fully Implicit Formulation of the Equations for Multiphase Flow in Porous Media, in *Multigrid Methods: Special Topics and Applications II*, GMD-Studien 189, pp. 31–42, 1990.

[7] Brandt, A., Guide to Multigrid Development, in *Multigrid Methods, Lecture Notes in Mathematics*, Hackbusch, W. and Trottenberg, U., editors, pp. 220–312, Springer-Verlag Berlin, 1982.

[8] Molenaar, J., A Simple Multigrid Method for 3D Interface Problems, Technical Report 94-44, TU Delft, 1994.

[9] Douglas Jr., J., Peaceman, D., and Rachford, H., A Method for Calculating Multi-Dimensional Immiscible Displacement, *Trans. AIME*, 216:297–306, 1959.

Coarsening Strategies for Unstructured Multigrid Techniques with Application to Anisotropic Problems

E. Morano D. J. Mavriplis V. Venkatakrishnan *

Institute for Computer Applications in Science and Engineering
MS 132C, NASA Langley Research Center
Hampton, VA 23681-0001 USA

ABSTRACT

Over the years, multigrid has been demonstrated as an efficient technique for solving inviscid flow problems. However, for viscous flows, convergence rates often degrade. This is generally due to the required use of stretched meshes (i.e. the aspect-ratio $AR = \Delta y/\Delta x << 1$) in order to capture the boundary layer near the body. Usual techniques for generating a sequence of grids that produce proper convergence rates on isotropic meshes are not adequate for stretched meshes. This work focuses on the solution of Laplace's equation, discretized through a Galerkin finite-element formulation on unstructured stretched triangular meshes. A coarsening strategy is proposed and results are discussed.

Introduction

Multigrid method has been shown to be successful for solving elliptic problems. This is mainly due to its good damping properties which result from two very simple principles. A usual Fourier analysis demonstrates that most of the commonly used solvers effectively damp the high frequencies of a signal. A low frequency component of a given signal on a fine mesh becomes a high frequency on a coarser one, hence the idea of solving the same problem on a sequence of meshes where all frequencies can be damped equally and, if enough grids are available, only a few iterations will be required to produce a converged solution (for more details see [1]). Despite these rather simple considerations, the multigrid algorithm is complex and difficult to implement. One of the difficulties resides in the generation of the sequence of grids for unstructured meshes. The convergence properties of the multigrid method depend upon the "quality" of these grids.

A sequence of meshes may be produced through two different methods. First, starting from a mesh that is not too fine but correctly represents the problem, finer meshes may be generated through refinement. A global refinement, performed through local subdivision of the triangles of the discretization, tends to preserve the geometrical features required to obtain an efficient multigrid method. However, this will clearly not be efficient in terms of computational cost, hence the local refinement technique where specific regions of the mesh are refined and then possibly adapted [2]. Although this method seems more reasonable, it

*This research was supported under the NASA contract No. NAS1-19480 while the authors were in residence at ICASE.

increases the computational time and the complexity of the multigrid algorithm. Another method consists in coarsening an existing fine mesh, which has been created to represent accurately the different phenomena to be observed. One of the techniques available consists in removing, through a coarsening criterion, a certain number of nodes from the initial mesh and to reconnect (retriangulate) the remaining set of nodes. This method is especially effective in the case of non stretched meshes [3]. The reconnection usually relies on the Delaunay technique [4] that tends to produce the "most equilateral" triangulation for the given point distribution and therefore is not easily applicable to stretched meshes. In order to avoid retriangulation, the so-called agglomeration technique (see Lallemand et al. [5]) is interesting. The generation of coarser meshes consists in the agglomeration, or fusion, of the control volumes of the discretization. However, for consistency considerations, when it comes to viscous flows, more accurate intergrid transfer operators are required [6, 7].

The following study focuses on the 2D Laplace's equation $\Delta u(x,y) = 0$, since the poor convergence properties of the multigrid technique, observed when solving the Navier-Stokes equations on stretched meshes, also appear for the solution of this simpler equation. The purpose of this work is to propose new coarsening strategies that will preserve the convergence rate of the usual isotropic multigrid technique. This is defined as a semi-coarsening method. This study will show how this process may be extended from the case of regular structured grids to totally unstructured meshes.

The organization of the paper is as follows: the discretization of the 2D Laplace's equation is introduced in Section 1 along with an edge-based data structure. Section 2 recalls the essential multigrid convergence properties. The generation of stretched grids is addressed in Section 3. A semi-coarsening algorithm, extended to unstructured meshes, is presented in Section 4. Finally, numerous experiments are discussed in Section 5.

1 Laplace's equation

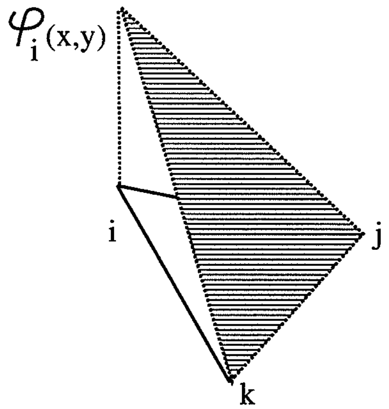


Figure 1: Linear basis function φ_i.

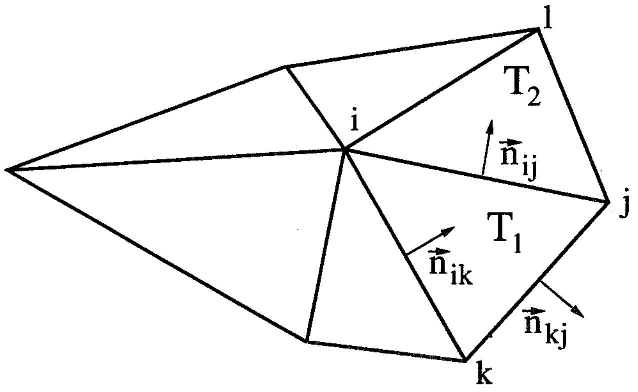


Figure 2: Vertex i and connecting neighbors.

The problem consists in solving Laplace's equation:

$$\begin{cases} \Delta u(x,y) = 0 \text{ on } \Omega \text{ convex polygonal domain.} \\ u = u_0 \text{ on } \Gamma. \end{cases} \tag{1}$$

A Galerkin Finite-Element formulation is used on unstructured triangular meshes. An integration by parts results in:

$$\int_{\Omega_i} \Delta u \; \varphi_i \; d\omega \;=\; -\int_{\Omega_i} \vec{\nabla} u \cdot \vec{\nabla} \varphi_i \; d\omega + \int_{\Gamma_i} \vec{\nabla} u \cdot \vec{n} \; \varphi_i \; d\sigma \tag{2}$$

where φ_i is the linear basis function as depicted in Fig.1. If u is piecewise linear, then the Green formula and the notations of Fig.2 result in:

$$\left\{ \begin{array}{lcl} (\vec{\nabla}\varphi_i)_{T_1} & = & \dfrac{-1}{2A_1}\, \vec{n}_{kj} \\ (\vec{\nabla}u)_{T_1} & = & \dfrac{-1}{2A_1}\, (u_i\vec{n}_{kj} + u_k\vec{n}_{ij} - u_j\vec{n}_{ik}) \end{array} \right. \tag{3}$$

where u_i is the value of the solution u on vertex i, A_1 is the area of triangle T_1, $\vec{n}_{ij}$ the vector normal to the edge $[i,j]$ and of magnitude equal to the length of the edge. Equation (2) can be rewritten as:

$$\int_{\Omega_i} \Delta u \; \varphi_i \; d\omega \;=\; \sum_i \int_{T_i} (\vec{\nabla}\varphi_i)_{T_i} \cdot (\vec{\nabla}u)_{T_i} \; d\omega \;=\; \sum_i \frac{1}{2A_i} \vec{n}_{kj} \cdot (\vec{\nabla}u)_{T_i} \tag{4}$$

Moreover, for the considered triangle T_1, (3) can be rewritten as:

$$\left\{ \begin{array}{lcl} (u_x)_{T_1} & = & \dfrac{1}{2A_1}\, (\Delta u_{ij}\Delta y_{jk} - \Delta u_{jk}\Delta y_{ji}) \\ (u_y)_{T_1} & = & \dfrac{-1}{2A_1}\, (\Delta u_{ij}\Delta x_{jk} - \Delta u_{jk}\Delta x_{ji}) \end{array} \right. \tag{5}$$

where $\Delta u_{ij} = u_j - u_i$. A similar formulation can be written for triangle T_2. In evaluating the coefficient for the edge joining vertices i and j, only the triangles T_1 and T_2 will yield non-zero contributions. The final expression of (4) is thus an edge-based formulation:

$$\int_{\Omega_i} \Delta u \; \varphi_i \; d\omega \;=\; \frac{1}{4} \sum_{edges} \left[\left(\frac{\Delta y_{ik}\Delta y_{jk}}{A_1} + \frac{\Delta y_{il}\Delta y_{jl}}{A_2} \right) + \left(\frac{\Delta x_{ik}\Delta x_{jk}}{A_1} + \frac{\Delta x_{il}\Delta x_{jl}}{A_2} \right) \right] \Delta u_{ij} \tag{6}$$

where the sum is taken over all incoming edges for vertex i. The geometrical anisotropy is reflected in the coefficient associated with each edge. If the length $\|\vec{ij}\|$ increases (the nodes k and l being fixed) then the value of the coefficient decreases. Therefore, considering the domain $\Omega_i = \bigcup_i T_i$, the maximum coefficient is associated with the smallest connecting edge and the minimum with the longest.

2 Some definitions and convergence results

Multigrid theory relies on the use of a sequence of nested meshes for solving (1). These meshes represent the different spaces where the equation is discretized. In what follows, only two meshes are considered: $\mathcal{H}_h$ and $\mathcal{H}_H$ with $H = 2h$ and $\mathcal{H}_H \subset \mathcal{H}_h \subset H_1^0$. The discrete problem on the fine grid is written as:

$$A_h u_h \;=\; 0 \tag{7}$$

A weighted Jacobi relaxation is considered as the basic iterative process or smoother:

$$u_h^{n+1} \;=\; S_h\, u_h^n \;=\; (I - \omega\, D_h^{-1}\, A_h)\, u_h^n, \text{ where } D_h \;=\; (A_h)_{ii} \tag{8}$$

In order to use both spaces for solving (7) it is necessary to use transfer operators. A linear interpolation $P :\; \mathcal{H}_H \longrightarrow \mathcal{H}_h$ defines the prolongation operator, and its transpose $R = P^* :\; \mathcal{H}_h \longrightarrow \mathcal{H}_H$ defines the restriction. The 2-Grid iterative operator M_h is then defined by:

$$\begin{array}{lcl} u_h^{n+1} \;=\; M_h\, u_h^n & = & S_h^{\nu_2}\, (I - P\, A_H^{-1}\, R\, A_h)\, S_h^{\nu_1}\, u_h^n \\ & = & (A_h^{-1} - P A_H^{-1} R)\, (A_h S_h^{\nu})\, u_h^n \end{array} \tag{9}$$

with $\nu_1 = \nu$ pre-relaxations and $\nu_2 = 0$ post-relaxations.

One very important feature of a multigrid (MG) algorithm is its mesh-independent convergence. According to Hackbush [8], mesh-independence for elliptic operators, is achieved through the smoothing property ($\|A_h S_h^{\nu}\| \leq h^{-2}\ \eta(\nu)$, where $\lim_{\nu \to \infty} \eta(\nu) = 0$) and the approximation property ($\|A_h^{-1} - PA_H^{-1}R\| = O(h^2)$). Because of its nature, the MG algorithm converges linearly with respect to the number of MG-cycles.

Morano et al., in [3], showed that this may also be achieved for the Euler and low Reynolds number Navier-Stokes equations where the employed meshes are not stretched. However, when highly-stretched elements are used (mandatory for high Reynolds number solutions, see [7] for example), this convergence greatly deteriorates with classical fully-coarsened (FC) grids. It is no longer linear nor mesh-independent. The deterioration in convergence is also observed when the resolution of Laplace's equation is attempted with highly stretched elements, that is, when the mesh is anisotropic.

3 A sequence of grids

When very stretched elements are used, the damping properties of the smoother are negligible in the stretching direction. Thus, using a full-coarsening strategy will certainly not improve the damping properties, since the stretching is fully preserved on larger elements. Moreover, the distribution of nodes in the stretching direction will correctly represent the low frequencies of the signal, whereas, in the direction normal to the stretching, it will represent the high frequencies. Because of the nature of the smoothers commonly used, the multigrid technique damps mainly the high frequencies, hence the idea of semi-coarsening in the direction normal to the stretching.

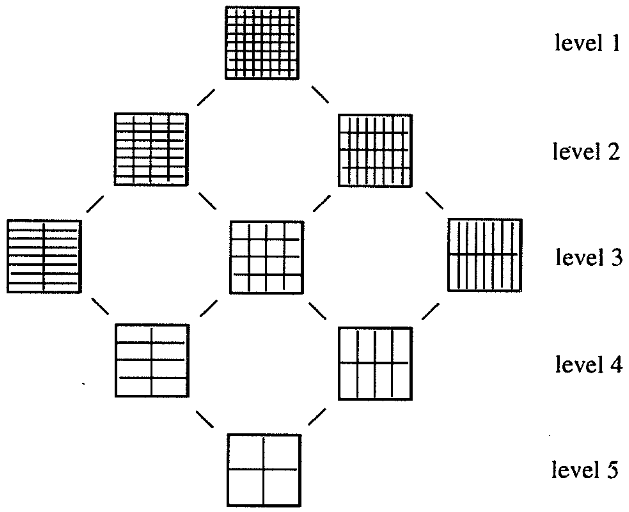
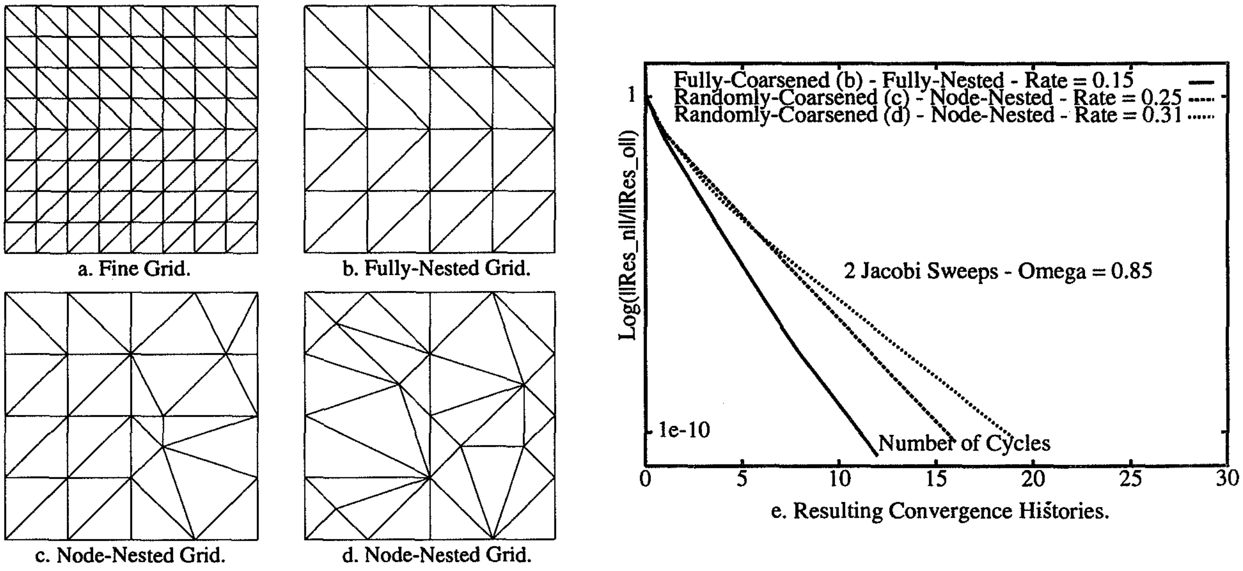


Figure 3: Sequence of grids for MSG.

The semi-coarsening technique is well known and used especially in the structured mesh community. For complex geometries, however, multiple directions within the mesh require semi-coarsening. A process named Multiple Semicoarsened Grid (MSG) Algorithm was introduced by Mulder [9]. This technique relies on the generation of numerous grids that are semi-coarsened (SC) from the finer grid in all possible directions as depicted in Fig.3. This ensures proper dissipation of the signal. A multigrid scheme is then implemented using all the grids which is complex and costly, especially for 3D problems [10]. Moreover, there is no possible extension of this technique to unstructured grids.

The complexity of the usual multigrid technique also relies on the full-coarsening method. This technique consists in removing every second vertex in each direction on a regular structured mesh, which results in a number of nodes of the coarse grid decreased by a factor 4. The V-cycle complexity of such a method tends to 4/3 WUs (a Work Unit corresponds to the computation of one residual on the fine grid). The semi-coarsening technique produces coarse grids with a number of nodes decreasing by a factor 2 and the overall complexity tends to 2. Therefore, such a method will cost more per cycle. However, it will be shown that this technique allows a much better damping factor than a regular full-coarsening technique in the case of stretched meshes.

The smoothing property is valid for the weighted Jacobi relaxation scheme applied in this study. The effect of the approximation property is emphasized since it determines the mesh-independence of the convergence. This property is verified when the discretized subspaces, defined by the sequence of coarser meshes, utilized within the MG algorithm are nested. In this paper, the sequence of meshes is created through a semi-coarsening technique followed by a retriangulation. When this strategy is applied to unstructured meshes, the nestedness of the meshes is rather difficult to preserve. The nodes of the coarse grid form a subset of the nodes of the fine grid which produces node-nested, but not element-nested, grids.


Fully-Coarsened (b) - Fully-Nested - Rate = 0.15
Randomly-Coarsened (c) - Node-Nested - Rate = 0.25
Randomly-Coarsened (d) - Node-Nested - Rate = 0.31
2 Jacobi Sweeps - Omega = 0.85
Log(||Res_n||/||Res_o||)
1
1e-10
Number of Cycles
0
5
10
15
20
25
30

a. Fine Grid.

b. Fully-Nested Grid.

c. Node-Nested Grid.

d. Node-Nested Grid.

e. Resulting Convergence Histories.

Figure 4: Coarse grid discretizations $AR = 1$.

The example depicted in Fig.4 shows how the convergence varies with respect to the nestedness of the meshes. A non-stretched 89 node Cartesian mesh defines the fine grid (Fig.4.a). The boundary conditions are those defined in Section 5. Three different coarse grids are considered. Each of them is a node-nested grid and comprises 25 nodes. Fig.4.b shows a usual fully nested grid. Fig.4.c and d depict randomly coarsened grids. On the right side of the grid shown in Fig.4.c a few elements are not nested. Finally, Fig.4.d depicts a coarsened grid where the elements are anything but nested. Two-grid experiments (see Section 5.1) are performed and Fig.4.e depicts the respective convergence histories. The convergence rate ranges from 0.15 to 0.31 for such a simple test-case. Therefore, the nestedness of the grids is of extreme importance in the quality of the MG performance. Further results may be obtained in [11].

4 Semi-coarsening and unstructured meshes

In what follows is presented a semi-coarsening technique that is applicable to unstructured meshes as well as to structured meshes. The technique may be seen as a variant of the Algebraic Multigrid (see [12]) in the sense that it necessitates a pre-processing stage that relies on the discretization of the equation for generating the coarse grids. As mentioned previously, the Galerkin discretization of Laplace's equation amounts to a sum over edges. The value of the coefficient associated with each edge is determined by the geometry of the surrounding elements (triangles). The smaller the length of the edge, the larger the value of the coefficient.

The semi-coarsening technique proceeds as follows: once a node is selected to remain on the coarse grid, its neighbors must be scanned to determine which one of them has to be removed. The removed node corresponds to the edge associated with the largest coefficient. The algorithm is two-fold. First, it has to go through the mesh and select the nodes to remain on the coarse grid, and, second, for each selected node, it has to determine which of its neighbors is to be removed. The setup employed for coarsening is the same as that used for agglomeration in [13, 7].

Unstructured meshes for high-Reynolds number flow computations are essentially comprised of two regions: one where the aspect-ratio is (very) small, where the viscous effects are dominant, and another one, where the aspect-ratio is close to 1, far from the viscous effects (the farfield for example). In order to preserve the low complexity of an MG algorithm it might be desirable to perform the semi-coarsening only in the low aspect-ratio region, whereas a full-coarsening may be applied elsewhere. Again, this is similar to an Algebraic Multigrid as described in [12]. This should provide a slightly better complexity than the one obtained through semi-coarsening only. The algorithm is written as:

1. For each node i on the fine grid the average and maximum values of the coefficients $coef_i$ of its connecting edges are computed: avg_i and max_i.
2. The parameter $\beta = \frac{1}{N}\sum_{i=1}^{N}\frac{max_i}{avg_i}$ provides an indication of the anisotropy.
3. The determination, through a heaplist, of the vertex *jpick* that remains on the coarse grid is then performed.
4. The removal of the connecting neighbor(s) of *jpick* is achieved through a coarsening criterion.
5. Goto [3].

The heaplist serves as an advancing front. The starting point of the front will determine the quality of the subset of nodes which constitute the coarse grid. Since semi-coarsening consists in removing every second vertex in the direction normal to the stretching, it is expected that the advancing front should be initiated from the region comprising the lowest aspect-ratio elements (the surface of an airfoil for example). Therefore, the following items are incorporated:

- Technical programming considerations make the front start first with the boundaries.
- The body and farfield extrema are retained on the coarse grid in order to preserve the general geometry of the discretized domain.
- The heaplist is determined by a "key-function" [14]. This "key-function" is defined by the connecting distance (minimum number of edges) to the boundary (or region where the aspect-ratio is minimum) of the unprocessed vertex (not in the front). The result is a list of edges where the first edge is associated with the minimum distance and *jpick* is its unprocessed vertex.

Once a node is selected to remain on the coarse grid, a semi-coarsening criterion determines which of the nb_{edge} connecting neighbors of *jpick* is to be removed:

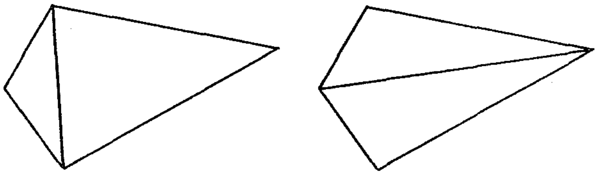
1. nb_{max} is defined by the maximum number of nodes to be deleted:
   if $max_{jpick} \geq \beta\ avg_{jpick}$ then $nb_{max} = 1$ (Semi-Coarsening),
   else $nb_{max} = nb_{edge}$ (Full-Coarsening).
2. The array $List_{jpick}$ contains the available unprocessed neighbors.
   n_{del}, the number of deleted nodes, is set equal to 0.

3. The determination of the available local maximum coefficient is performed: $loc_{max} = \max_{i \in List_{jpick}} (coef_i)$.

4. A node $i \in List_{jpick}$ is removed if: $coef_i = loc_{max}$ and $loc_{max} \geq avg_{jpick}$. That is if its value is equal to the maximum local coefficient and if this maximum is greater than the average value of all the surrounding coefficients.

5. The array $List_{jpick}$ is updated along with the number of deleted nodes ($n_{del} \leftarrow n_{del} + 1$). If $n_{del} < nb_{max}$ goto [3].

This algorithm clearly provides a semi/full-coarsening (S/FC) technique. Yet, if appropriate, the algorithm only performs semi-coarsening or full-coarsening. Such an algorithm may be applied to unstructured meshes as well as to structured meshes provided the considered discretization relies on an edge-based data structure. This algorithm relies on the discretization of the equation to be solved rather than on simple geometrical considerations.



a. Delaunay - Max Min.   b. Min Max - Variant.

Figure 5: Retriangulation techniques.

Once the subset of nodes of the fine grid is obtained after coarsening, it needs to be retriangulated. The reconnection relies here on a Delaunay method. This method has proved useful and efficient when used in conjunction with equilateral triangle types of meshes. The coarsening technique utilizing such an algorithm was introduced in [15]. Unfortunately, this method does not apply to highly stretched meshes. It usually results in a poor reconnection in the region where the nodes of the mesh are not regularly distributed. In order to overcome this difficulty, an edge-swapping technique may be employed [16, 17]. The Delaunay reconnection of a set of four nodes results in two triangles where the minimum angle is maximized (Fig.5.a). In lieu of preserving this connectivity it is possible to swap the edges by minimizing the maximum angle of the two triangles (Fig.5.b). This technique has proved very efficient when used with an advancing front technique for generating meshes, and is thus employed for the unstructured test-case in this paper. The reconnection of the structured coarse grids is performed through the usual Delaunay method.

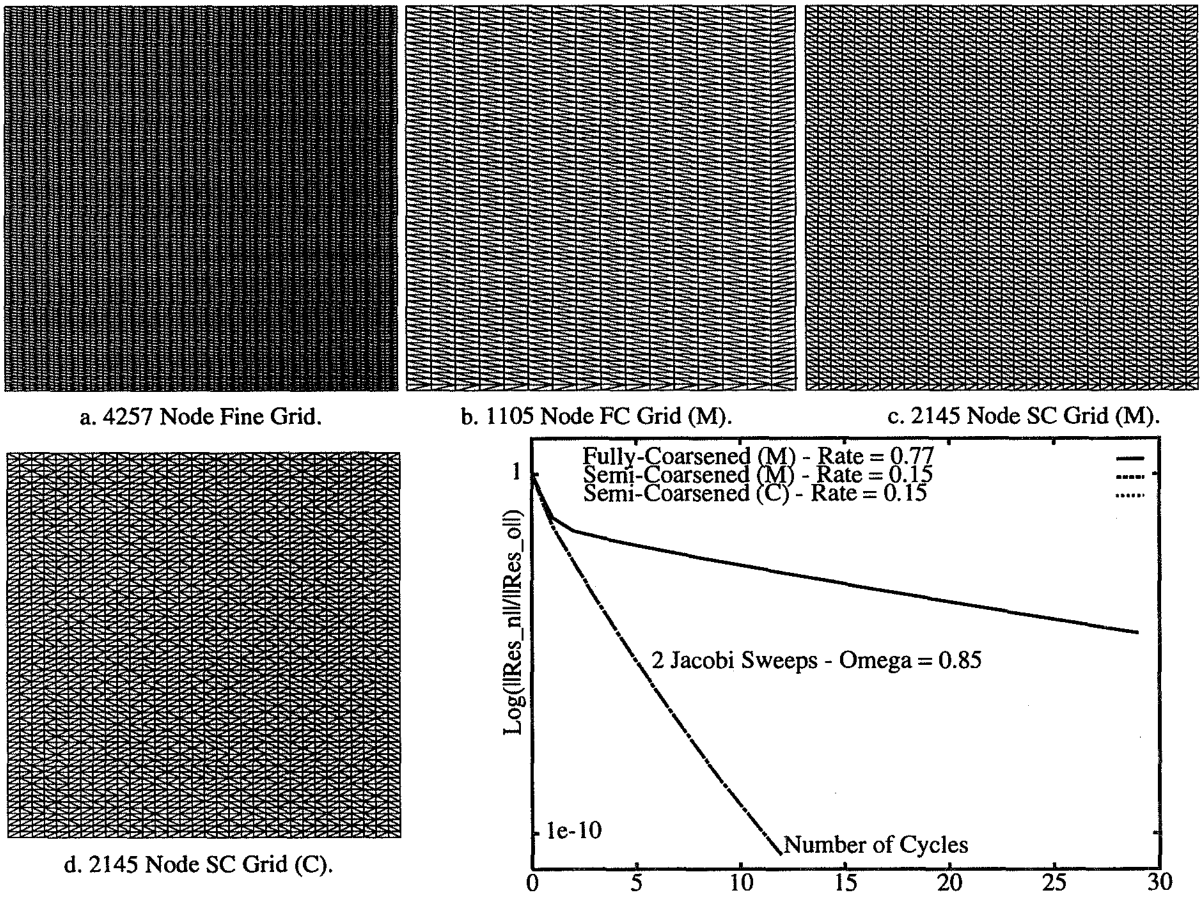
5 Results and comments

In order to validate the previous concept, various test-cases are performed for solving the Laplace's equation. Results are presented on structured and unstructured meshes. The discretization domain for the structured cases is defined by a square of surface 1, while the unstructured case is defined by a pentagon plunged in an unstructured mesh. A non-stretched structured test-case serves as the standard test-case since it provides the best MG convergence. The relaxation parameter ω is equal to 0.85 and no optimization is performed here. Two sweeps are performed on the fine grid. The transfer operators are linear and were introduced in [18]. All cases are performed with Dirichlet boundary conditions. For the structured test-cases they are defined by $u(0,x) = 1$, $u(x,1) = 2$, $u(1,x) = 3$ and $u(x,0) = 4$, and for the unstructured case they are equal to -1 on the body and to 1 on the farfield. For all test-cases, the different grids used are presented along with the convergence histories of the various schemes. The convergence histories depict the logarithm of the norm of the normalized residual with respect to the number of cycles. This convergence is carried over until a residual decrease on the fine grid equal to 10^{-10}.

5.1 Two-Grid experiments

These experiments require a residual decrease on the coarse grid equal to 10^{-10}. The semi-coarsening-only ($nb_{max} = 1$) option of the algorithm is used for the generation of the coarse grids.

Non-stretched Meshes. The aspect-ratio is equal to one and the grids are fully-nested. The fine and coarse grid, respectively, are similar to those depicted in Fig.4.a and b with 4225 (65 x 65) and 1089 (33 x 33) nodes, respectively . The coarse-grid is a manually (M) fully-coarsened grid (i.e. the coarsening algorithm is not involved). No anisotropy is encountered here and a solution is obtained after 12 cycles which corresponds to a convergence rate of 0.15.



a. 4257 Node Fine Grid.

b. 1105 Node FC Grid (M).

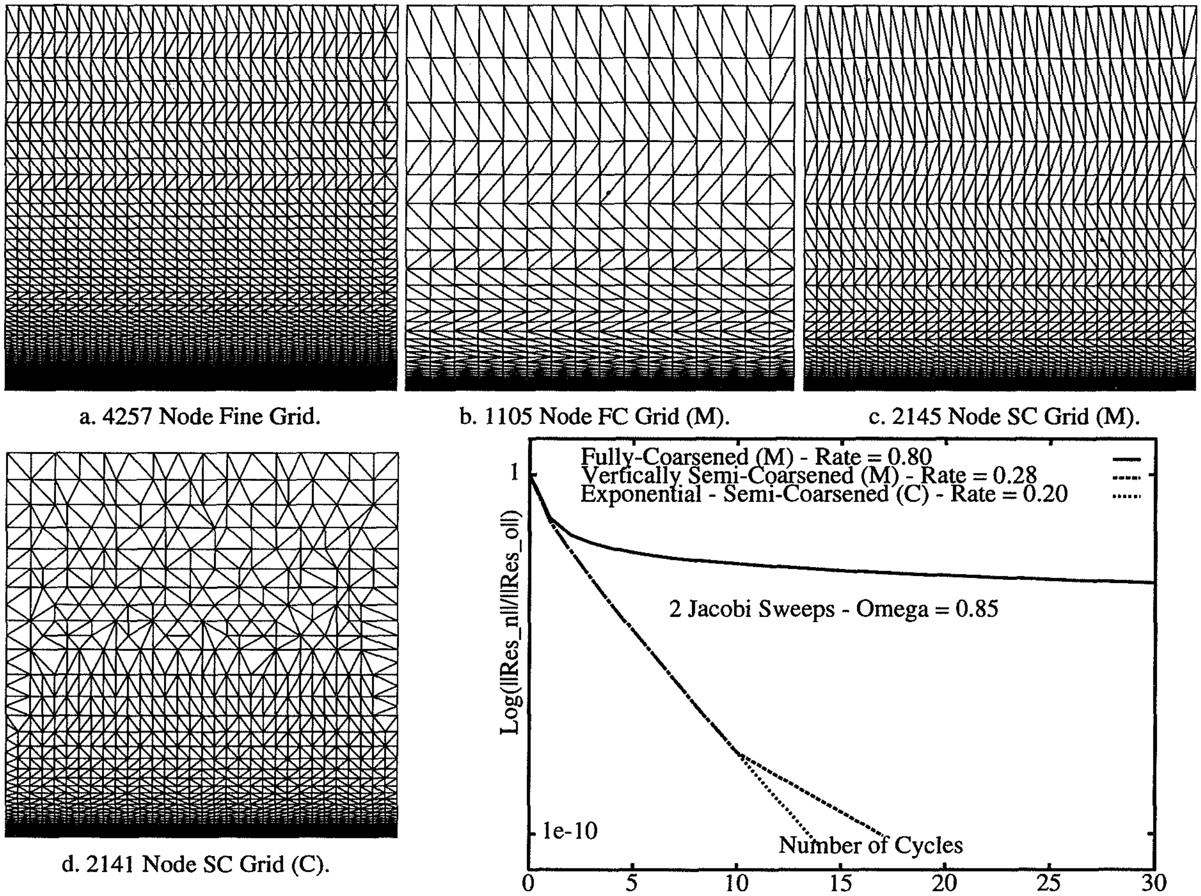
c. 2145 Node SC Grid (M).

d. 2145 Node SC Grid (C).

e. Resulting Convergence Histories.

Figure 6: Linear Meshes - $AR = 1/4$.

Linear Meshes. A 4257 (33 x 129) node fine grid is built (Fig.6.a) where the distribution of nodes is linear in the vertical (normal to the stretching) direction and the aspect-ratio is equal to 1/4. Three types of coarser meshes are presented. In Fig.6.b is depicted a manually fully-coarsened 1105 (17 x 65) node coarse grid, that represents the classical coarsening technique. In Fig.6.c and d are depicted two semi-coarsened grids. The first grid is obtained manually through a vertical semi-coarsening in a 2145 (33 x 65) node coarse grid. The second grid is the result of the coarsening algorithm (C) applied to the fine grid. It is a 2145 node coarse grid. The triangulations of the two semi-coarsened grids appear to be different while the subset of nodes are the same. Yet, similar convergences are expected. In Fig.6.e are depicted the various convergence histories. The full-coarsening technique results in a convergence rate of 0.77 while the semi-coarsening techniques provide both a convergence rate equal to 0.15, which is identical to the convergence rate of the non-stretched test-case.



a. 4257 Node Fine Grid.

b. 1105 Node FC Grid (M).

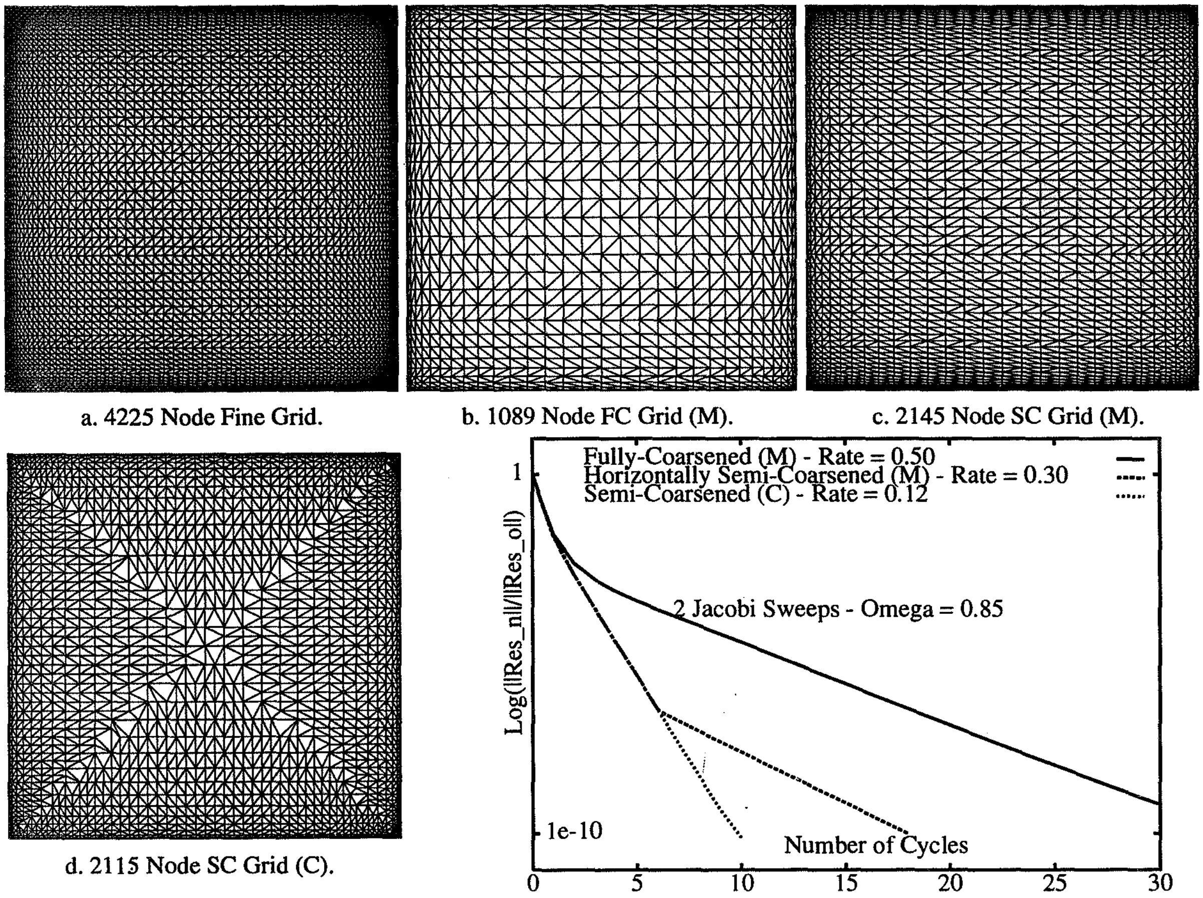
c. 2145 Node SC Grid (M).

d. 2141 Node SC Grid (C).

e. Resulting Convergence Histories.

Figure 7: Exponential Meshes - $AR = 2.4 \times 10^{-4}$.

Exponential Meshes. A 4257 (33 x 129) node fine grid is depicted in Fig.7.a. The distribution of nodes is exponential in the vertical direction. The minimum aspect-ratio is equal to 2.4×10^{-4} and the maximum to 2.2. This grid is manually fully-coarsened which produces a 1105 (17 x 65) node coarse grid (Fig.7.b). A manually vertically semi-coarsened 2145 (33 x 65) node coarse grid is depicted in Fig.7.c. Where the stretching follows the horizontal direction (where the distribution of nodes is more dense) this technique will provide the expected result, while the stretching deteriorates in the vertical direction (where the distribution of nodes is less dense). A 2141 node coarse grid obtained with the coarsening algorithm is depicted in Fig.7.d. In this case the coarsening follows the direction normal to the stretching everywhere in the mesh, as can be seen in the less dense region. The full-coarsening technique results in a 0.80 convergence rate (Fig.7.e). The manually semi-coarsened grid proves to have a much better convergence rate of 0.28, but the best convergence rate of 0.20 corresponds to the automatically semi-coarsened grid. Moreover, the vertically semi-coarsened grid shows a change of slope at the end of the convergence. This means that the MG algorithm does not perform optimally and does not damp low frequencies correctly, whereas the code semi-coarsened grid provides a linear-type of convergence rate. Therefore, and although both semi-coarsened grids have similar numbers of nodes, the coarse grid obtained through the automated coarsening algorithm results in more optimal convergence.



a. 4225 Node Fine Grid.

b. 1089 Node FC Grid (M).

c. 2145 Node SC Grid (M).

d. 2115 Node SC Grid (C).

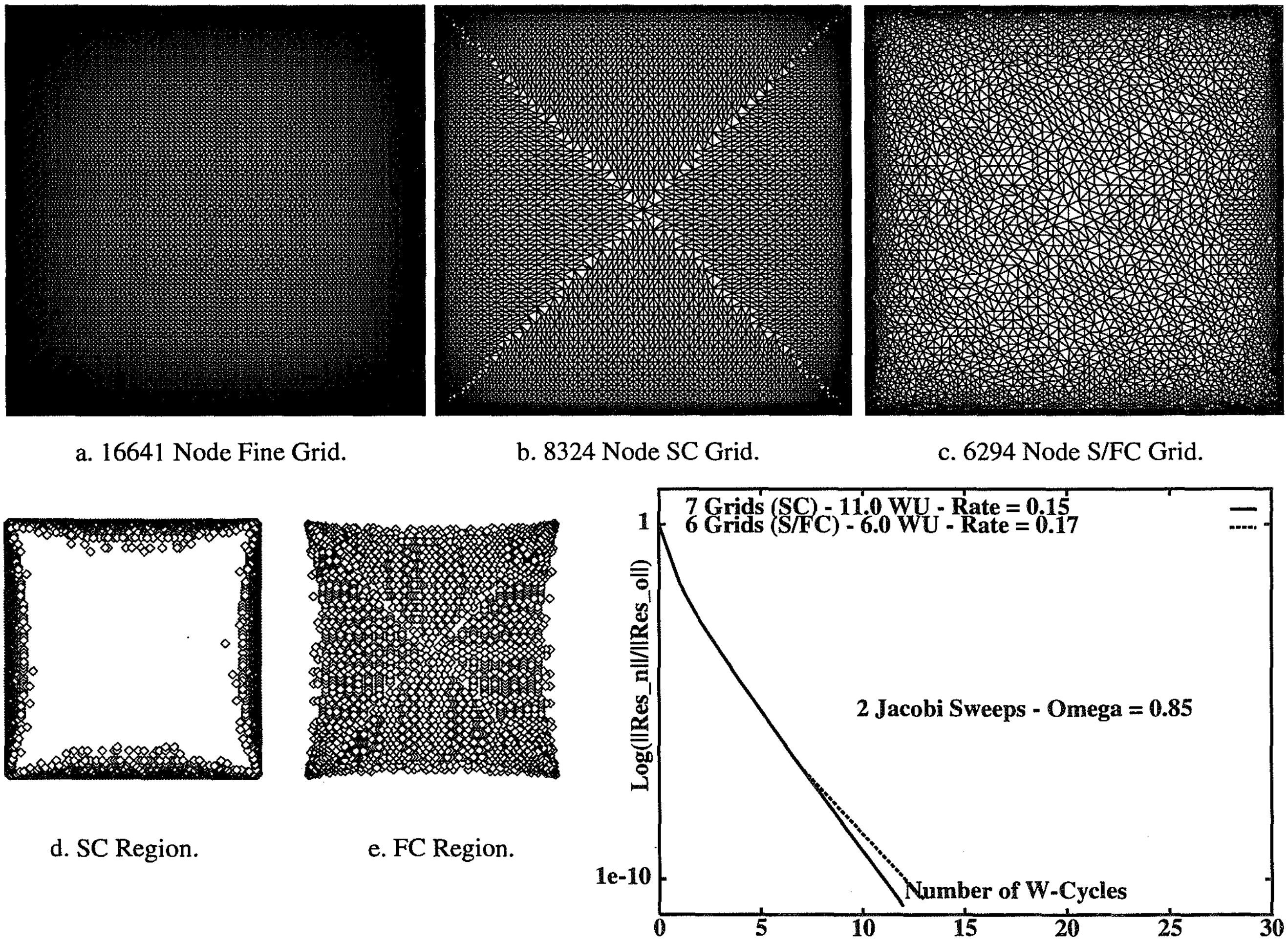
e. Resulting Convergence Histories.

Figure 8: Chebyshev Meshes - $AR = 0.024$.

Chebyshev Meshes. A 4225 (65 x 65) node fine grid is built where the distribution of nodes is a cosine function in both directions. The minimum aspect-ratio is equal to 0.024 and the maximum to 40.73 (Fig.8.a). This grid comprises stretched and non-stretched elements. The minimum aspect-ratio cells are essentially located on the boundary of the domain, while the maximum aspect-ratio cells are located in the bisectors and in the middle of the domain. A manually fully coarsened 1089 (33 x 33) node grid is depicted in Fig.8.b. Although no natural manual semi-coarsening technique applies here, a horizontally semi-coarsened 2145 node (33 x 65) coarse grid is built for comparison purposes (Fig.8.c). The coarsening algorithm resulted in a 2115 node coarse grid (Fig.8.d). It is again obvious that the semi-coarsening follows the direction normal to the stretching, each region being clearly separated by the bisectors. The fully-coarsened grid provided a convergence rate of 0.50, and 0.30 was achieved with the manually horizontally semi-coarsened grid (Fig.8.e). A linear type of convergence resulting in a convergence rate of 0.12 was achieved with the code semi-coarsened grid. It is interesting to note that, despite the similar number of nodes shared by the manually horizontally semi-coarsened grid and the code semi-coarsened grid, they provided different results, and therefore the good convergence rate of the code semi-coarsening technique cannot be attributed solely to the number of nodes on the coarse grid.

5.2 Multigrid experiments

In this section, multigrid experiments are explored in order to demonstrate the robustness of the algorithm in producing a sequence of grids that permit efficient MG convergence. The number of grids will vary according to the test-case. Two sweeps of the Jacobi relaxation are performed on each level and W-cycles are employed since they provide a better resolution of the coarse grid, resulting in better convergence rates. A structured Chebyshev and an unstructured test-case are performed with both semi and semi/full-coarsening techniques.



a. 16641 Node Fine Grid.

b. 8324 Node SC Grid.

c. 6294 Node S/FC Grid.

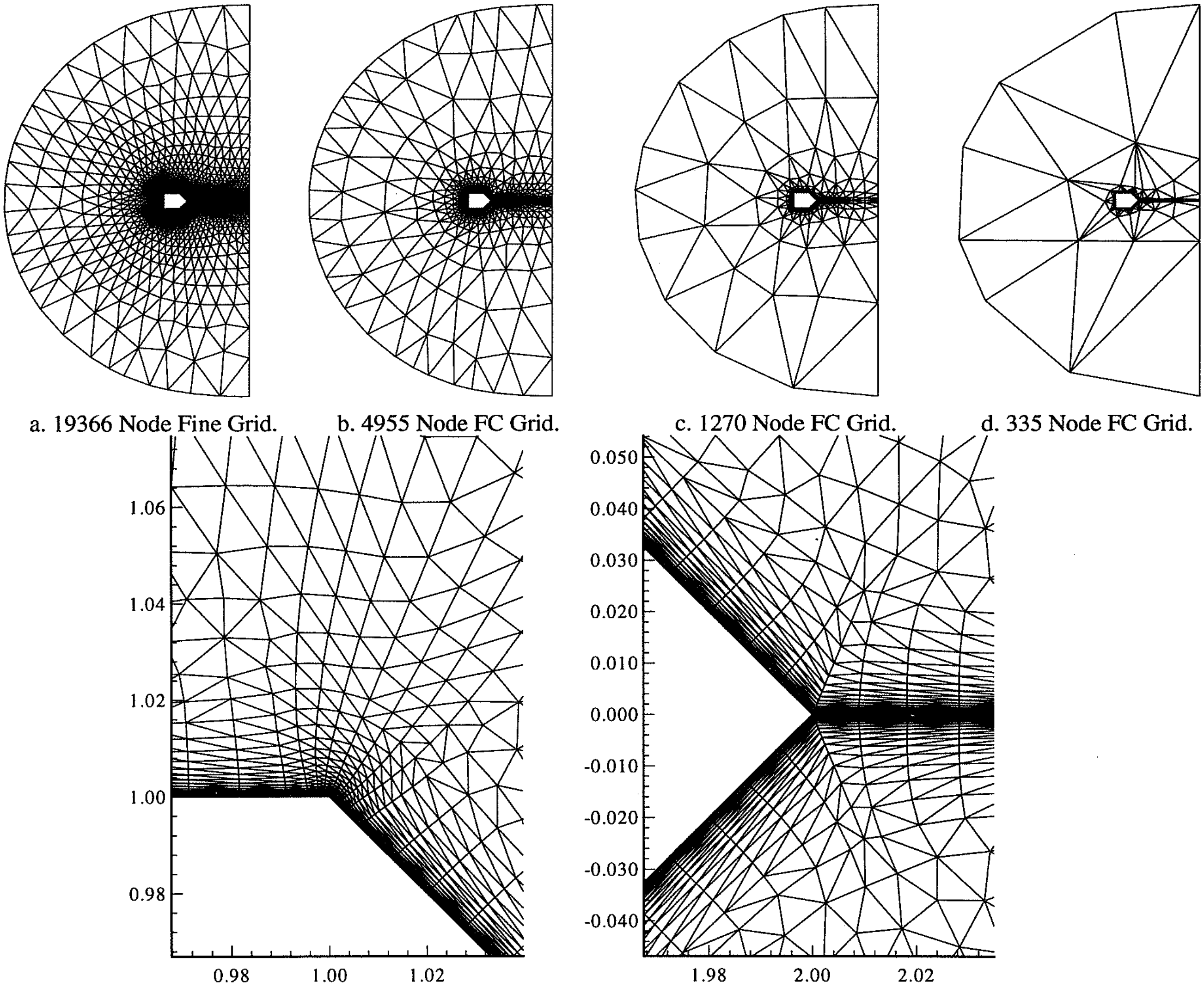
d. SC Region.

e. FC Region.

f. Resulting Convergence Histories.

Figure 9: Multigrid Chebyshev Meshes - $AR = 0.012$.

The Chebyshev test-case. A 16641 (129 x 129) node fine grid is constructed with a minimum aspect-ratio value of 0.012 and a maximum value of 81.50 (Fig.9.a). The semi-coarsening option provides a sequence of 7 grids comprising 16641, 8324 (shown Fig.9.b), 4329, 2289, 1211, 652 and 352 nodes, and the semi/full-coarsening technique a sequence of 6 grids comprising 16641, 6294 (shown Fig.9.c), 2976, 1077, 559 and 286 nodes. The respective W-cycle complexities are equal to 11 and 6 WUs. The region where the algorithm performs the semi-coarsening is depicted nodewise in Fig.9.d, while Fig.9.e shows where the full-coarsening is applied. It is clear that the semi-coarsening is applied to the highly stretched element region as expected. The semi-coarsening technique results in a standard-like convergence rate of 0.15 (Fig.9.f). When used only with 6 grids, this technique requires the coarsest grid to be converged completely, otherwise the process abruptly stalls at some low residual value. A convergence rate of 0.17 and a low complexity favor the semi/full-coarsening technique. Yet, the convergence history displays a (slight) change of slope. This indicates that the method is sensitive to the quality of the triangulation of the coarse grids. Mesh-independent convergence is the purpose of this study, and is only truly achieved with the semi-coarsening technique. The slightly poorer type of convergence associated with the semi/full-coarsening technique may be explained by the quality of the triangulation of the coarse grid. Full-coarsening in non-stretched regions tends to deteriorate the relative difference of aspect-ratio between the highly and non-stretched regions. Moreover, the addition of a 7th grid, or even converging the coarsest level, does not change the convergence.

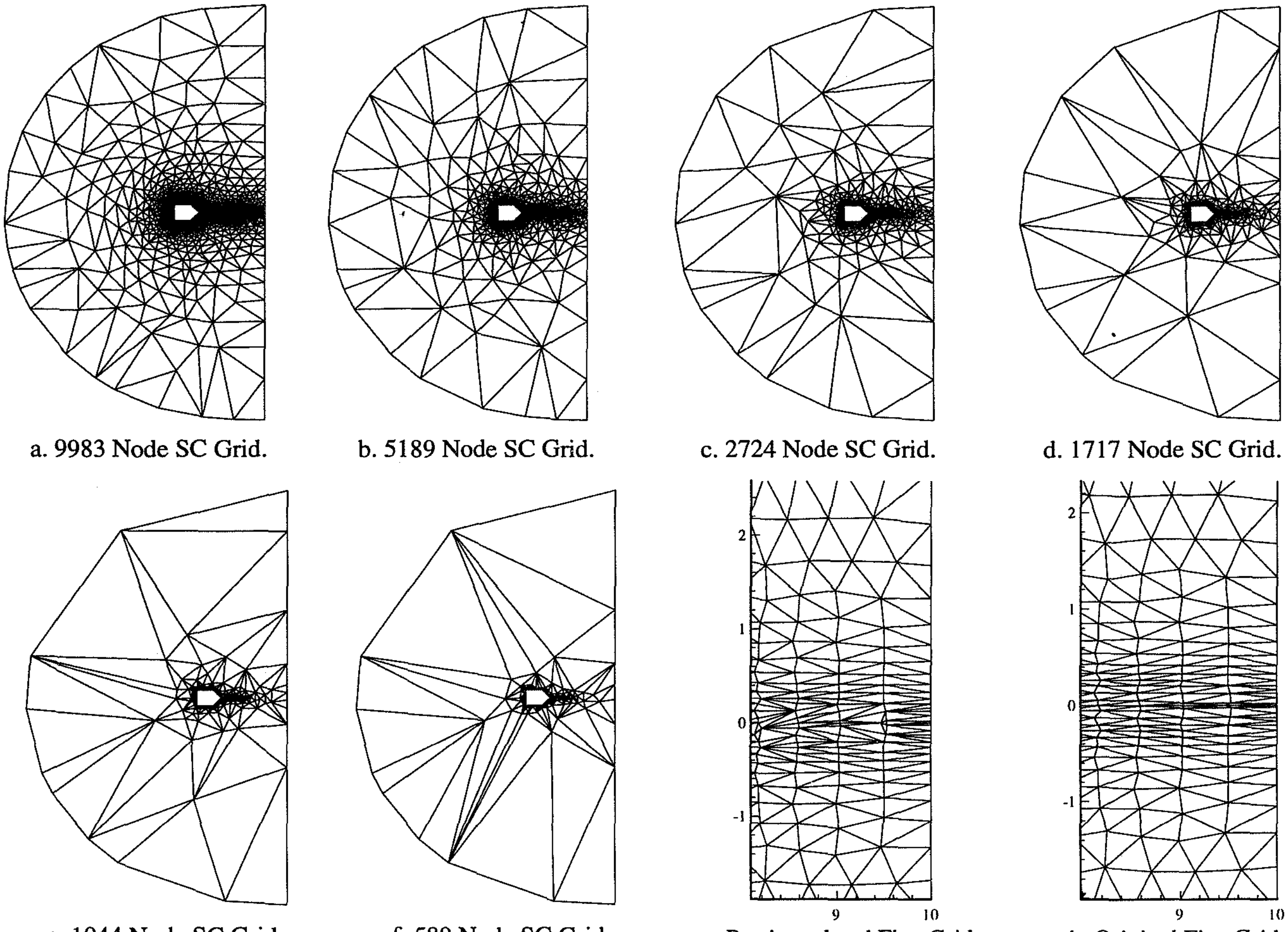


a. 19366 Node Fine Grid. b. 4955 Node FC Grid. c. 1270 Node FC Grid. d. 335 Node FC Grid.

e. Right Upper Corner. f. Wake Region.

Figure 10: Multigrid Unstructured - Full-Coarsening - $AR = 3.7 \times 10^{-5}$.

The unstructured test-case. In this case (Fig.10.a), a grid-spacing $\Delta y = 10^{-6}$ on the body results in an average minimum aspect-ratio of 3.7×10^{-5}. In Fig.10.e and f are depicted the zoom of the right upper corner and of the wake region respectively in order to show the different type of stretched and non-stretched elements that appear in these meshes. A first sequence of 4 fully-coarsened meshes is manually constructed. The number of nodes for each level are: 19366, 4955, 1270 and 335. These meshes are depicted in Fig.10.a to Fig.10.d. The complexity of a W-cycle is equal to 3.2 WUs.

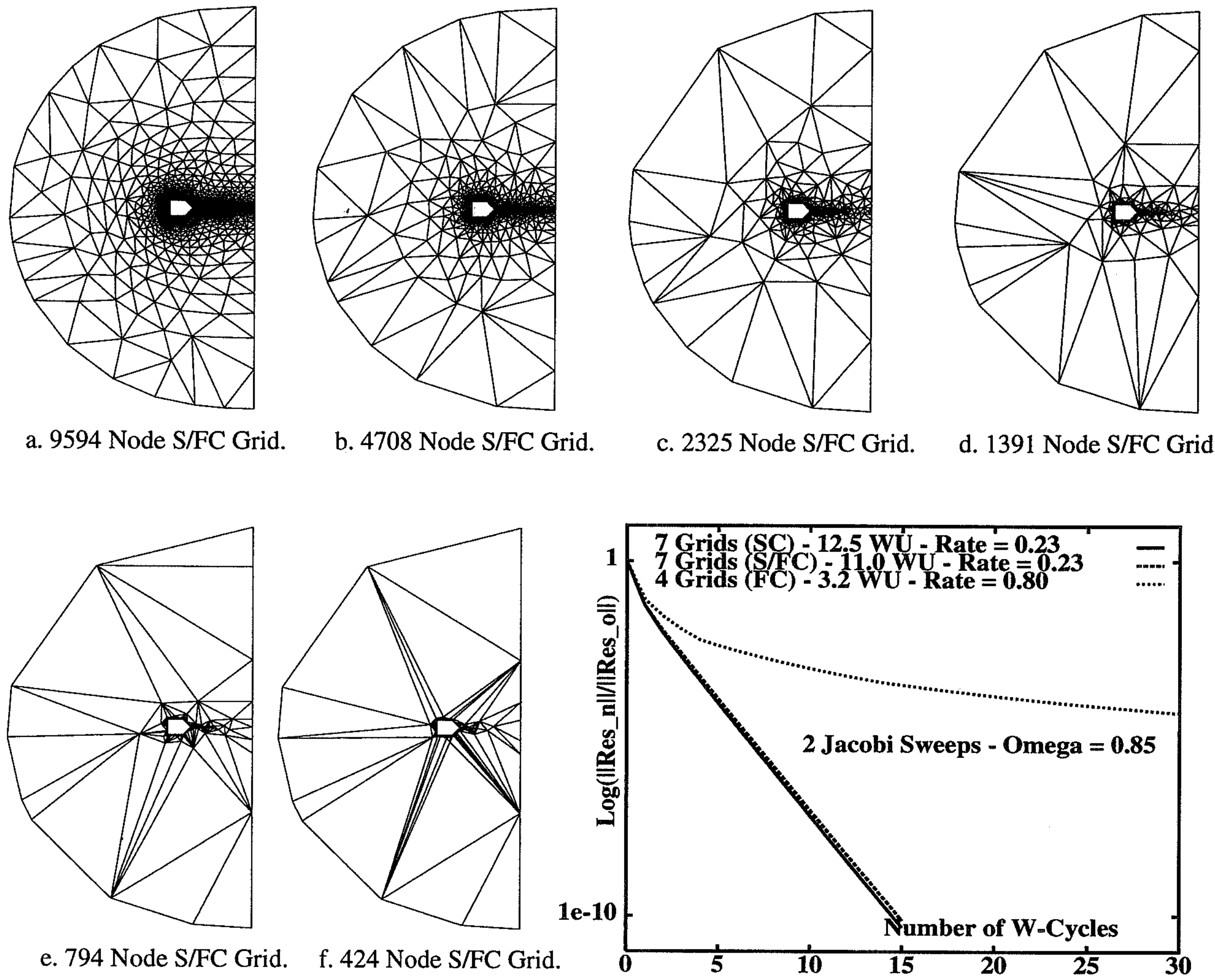


a. 9983 Node SC Grid. b. 5189 Node SC Grid. c. 2724 Node SC Grid. d. 1717 Node SC Grid.

e. 1044 Node SC Grid. f. 589 Node SC Grid. g. Retriangulated Fine Grid. h. Original Fine Grid.

Figure 11: Multigrid Unstructured - Semi-Coarsening - $AR = 3.7 \times 10^{-5}$.

The second sequence is obtained with the semi-coarsening technique only. There are 7 meshes that have 19366, 9983, 5189, 2724, 1717, 1044 and 589 nodes (Fig.11.a to Fig.11.f). The W-cycle complexity is equal to 12.5 WUs. The last sequence of meshes results from the semi/full-coarsening technique and provides 7 meshes (Fig.12.a to Fig.12.e): they comprise 19366, 9594, 4708, 2325, 1391, 794 and 424 nodes, resulting in a 11 WU W-cycle complexity. SC and S/FC methods required all coarse point sets to be retriangulated using the Min-Max Delaunay variant. In order to maintain favorable convergence rates, it was found that the fine grid needed to be retriangulated according to the same technique. This can partially be explained by the quality of the nestedness of all the grids as seen in Section 3. The fine grid is not depicted here for these last two sequences because it would appear similar to the original (Fig.10.a). However, the difference between the original and retriangulated fine grids, mostly confined to wake regions, is illustrated in Fig.11.g and h.



a. 9594 Node S/FC Grid. b. 4708 Node S/FC Grid. c. 2325 Node S/FC Grid. d. 1391 Node S/FC Grid.

e. 794 Node S/FC Grid. f. 424 Node S/FC Grid.

f. Resulting Convergence Histories.

Figure 12: Multigrid Unstructured - Semi/Full-Coarsening - $AR = 3.7 \times 10^{-5}$.

Converging the coarsest grid of the sequence of the fully-coarsened grids does not change the convergence rates equal to 0.80 (Fig.12.f). This indicates that the use of an additional coarser grid would not change the convergence. Besides, the retriangulation of the entire sequence of the fully-coarsened grids does not change the convergence rate of the MG algorithm, whether or not the coarsest grid is converged. The semi/fully-coarsened and semi-coarsened grids provide a clear improvement with respect to the usual fully-coarsened grids with convergence rates equal to 0.23. The semi/fully-coarsened grids demonstrate a better behavior than in the Chebyshev case because they are very similar to the semi-coarsened grids. Indeed, since most of the nodes are concentrated in the highly stretched regions, the algorithm performs essentially as a semi-coarsening technique. This type of meshes is more similar to exponential-type meshes rather than Chebyshev meshes.

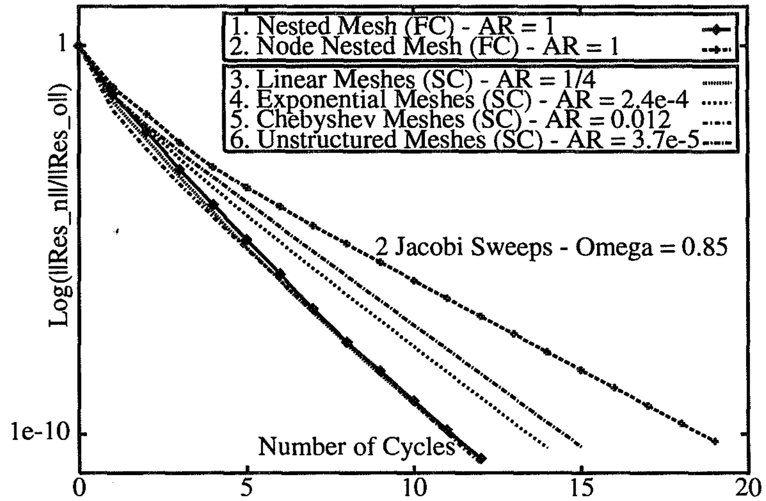


Figure 13: Significant Results.

Concluding remarks

In Fig.13 are gathered the most significant results. They are separated in two different subsets. Curves 1 and 2 represent the spectrum of convergences within which the other convergence histories must fit. Indeed, curve 1 shows the best convergence and curve 2 shows what is expected when the discretization subspaces are only node-nested. All other curves depict the convergence histories of the various test-cases that employ the semi-coarsening algorithm. The problem to be solved is the same for all test-cases, only the geometries of the discretized spaces differ. The results are straight lines with similar slopes that fall within the predicted range. The difference of slopes may be explained by two essential reasons. First, the boundary conditions of the structured and unstructured test-cases differ. It is not possible, due to the geometry, to transpose exactly the same boundary for both types. Then, it has been shown that the nestedness of the subspaces influences the quality of the convergence. It cannot be expected that the unstructured grids be completely nested. On the other hand the quality of the triangulation per grid may also damage the convergence.

In this paper, a new semi-coarsening algorithm relying on the discretization of the equation, which should enable flexible applications, has been introduced. Convergence rates for highly stretched unstructured meshes have been obtained similar to those for standard Cartesian structured non stretched meshes. Finally, linear, hence mesh independent, convergence rates have been demonstrated. The extension of these unstructured semi-coarsening techniques to the resolution of the Navier-Stokes equations is planned in the near future.

References

[1] W. Briggs. *A Multigrid Tutorial.* SIAM Philadelphia, 1987.

[2] K. Riemslagh and E. Dick. A multigrid method for steady Euler equations on unstructured adaptive grids. In *Sixth Copper Mountain Conference on Multigrid Methods*, pages 527–542. NASA, 1993. NASA Conference Publication 3224, Part 2.

[3] E. Morano and A. Dervieux. Looking for $O(N)$ Navier-Stokes solutions on non-structured meshes. In *Sixth Copper Mountain Conference on Multigrid Methods*, pages 449–463. NASA, 1993. NASA Conference Publication 3224, Part 2.

[4] H. Guillard. Node nested multigrid with Delaunay coarsening. *INRIA Research Report 92-12*, 1992.

[5] M.-H. Lallemand, H. Stève, and A. Dervieux. Unstructured multigridding by volume-agglomeration: Current status. *Computers and Fluids*, 21:397–433, 1992.

[6] B. Koobus, M.-H. Lallemand, and A. Dervieux. Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation. *The International Journal for Numerical Methods in Fluids*, 18:27–42, 1994.

[7] D. Mavriplis and V. Venkatakrishnan. Agglomeration multigrid for viscous turbulent flows. *AIAA Paper 94-2332*, 1994. To appear in Computers and Fluids.

[8] W. Hackbush. *Multigrid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[9] W. Mulder. A New Multigrid Approach to Convection Problems. *Journal of Computational Physics*, 83:303–329, 1989.

[10] A. Overman and J. Van Rosendale. Mapping robust parallel multigrid algorithms to scalable memory architecture. In *Sixth Copper Mountain Conference on Multigrid Methods*, pages 635–648. NASA, 1993. NASA Conference Publication 3224, Part 2.

[11] S. Zhang. Optimal-order nonnested multigrid methods for solving finite element equations 1: on quasi-uniform meshes. *Mathematics of Computation*, 55:23–36, 1990.

[12] J.W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid Methods*, pages 73–130. SIAM, Pennsylvania, S.F. McCormick Ed., 1987.

[13] V. Venkatakrishnan and D. Mavriplis. Agglomeration multigrid for the three dimensional Euler equations. *AIAA Paper 94-0069*, 1994. To appear in AIAA Journal.

[14] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* McGraw-Hill, 1992.

[15] E. Morano, H. Guillard, A. Dervieux, M.-P. Leclercq, and B. Stoufflet. Faster relaxations for non-structured mg with voronoï coarsening. In *First European Computational Fluid Dynamics Conference*, pages 69–74. Elsevier, 1992.

[16] T. J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. *AIAA Paper 91-0721*, 1991.

[17] D.L. Marcum and N.P. Weatherhill. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Paper 94-1926*, 1994.

[18] M.-P. Leclercq and B. Stoufflet. Characteristic Multigrid Method Application to solve the Euler equations with unstructured and unnested grids. In *International Conference on Hyperbolic Problems*, 1989.

PRECONDITIONING OPERATORS ON UNSTRUCTURED GRIDS

S.V. Nepomnyaschikh*

March 14, 1996

Abstract

We consider systems of mesh equations that approximate elliptic boundary value problems on arbitrary (unstructured) quasi-uniform triangulations and propose a method for constructing optimal preconditioning operators. The method is based upon two approaches: (1) the fictitious space method, i.e., the reduction of the original problem to a problem in an auxiliary (fictitious) space, and (2) the multilevel decomposition method, i.e., the construction of preconditioners by decomposing functions on hierarchical meshes. The convergence rate of the corresponding iterative process with the preconditioner obtained is independent of the mesh step. The preconditioner has an optimal computational cost: the number of arithmetic operations required for its implementation is proportional to the number of unknowns in the problem. The construction of the preconditioning operators for three dimensional problems can be done in the same way.

1 INTRODUCTION

Let $\Omega \subset \mathbb{R}^2$ be a domain with a piecewise smooth boundary Γ which belongs to the class C^2 and satisfies the Lipschitz condition [18]. In the domain Ω

*Computing Center, Siberian Branch Russian Academy of Sciences, 6 Lavrentiev av., Novosibirsk, 630090, Russia. The work was partially supported by the ISF under contract NPB 000, the grant DRET 93/34/401, the Russian Basic Research Foundation grant 93–01–01783

we consider the boundary value problem

$$-\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}\, a_{ij}(x)\, \frac{\partial u}{\partial x_j} + a_0(x)u = f(x)\,, \qquad x \in \Omega$$

$$u(x) = 0\,, \qquad x \in \Gamma_0 \tag{1}$$

$$\frac{\partial u}{\partial N} + \sigma(x)u = 0\,, \qquad x \in \Gamma_1$$

where

$$\frac{\partial u}{\partial N} = \sum_{i,j=1}^{2} a_{i,j}(x)\, \frac{\partial u}{\partial x_j}\, \cos(n, x_i)$$

is the conormal derivative, n denotes the outward normal to Γ, and Γ_0 is a union of a finite number of curvilinear segments, $\Gamma = \Gamma_0 \cup \Gamma_1$, $\Gamma_0 = \bar{\Gamma}_0$. Here $\bar{\Gamma}_0$ denotes the closure of Γ_0.

By $H^1(\Omega, \Gamma_0)$ we denote the subspace of the Sobolev space $H^1(\Omega)$

$$H^1(\Omega, \Gamma_0) = \{v \in H^1(\Omega) \mid v(x) = 0,\; x \in \Gamma_0\}\,.$$

We introduce a bilinear form $a(u, v)$ and a linear functional $l(v)$ as follows:

$$a(u, v) = \int_\Omega \Big(\sum_{i,j=1}^{n} a_{ij}(x)\, \frac{\partial u}{\partial x_j}\, \frac{\partial v}{\partial x_i} + a_0(x)uv \Big)\, \mathrm{d}x + \int_{\Gamma_1} \sigma(x)uv\, \mathrm{d}x$$

$$l(v) = \int_\Omega f(x)v\, \mathrm{d}x\,.$$

Let us suppose that the operator coefficients and the right-hand side of problem (1.1) are such that the bilinear form $a(u, v)$ is symmetric, elliptic and continuous on $H^1(\Omega, \Gamma_0) \times H^1(\Omega, \Gamma_0)$, i.e.,

$$a(u, v) = a(v; u) \qquad \forall u, v \in H^1(\Omega, \Gamma_0)$$

$$\alpha_0 \|u\|^2_{H^1(\Omega)} \le a(u, u) \le \alpha_1 \|u\|^2_{H^1(\Omega)} \qquad \forall u \in H^1(\Omega, \Gamma_0)$$

and the linear functional $l(v)$ is continuous on $H^1(\Omega, \Gamma_0)$:

$$|l(u)| \le \alpha \|u\|_{H^1(\Omega)} \qquad \forall u \in H^1(\Omega, \Gamma_0)\,.$$

The generalized solution $u \in H^1(\Omega, \Gamma_0)$ of problem (1.1) is, by definition, a solution to the projection problem [2]

$$u \in H^1(\Omega, \Gamma_0):\ a(u,v) = l(v) \qquad \forall v \in H^1(\Omega, \Gamma_0)\,. \tag{2}$$

It is familiar that under these assumptions concerning $a(u,v)$ and $l(v)$ there exists a unique solution of problem (1.2).

Let a positive parameter h be fixed (we always suppose that h is sufficiently small). Let

$$\Omega^h = \bigcup_{i=1}^{M} \tau_i$$

be a triangulation of the domain Ω (Ω^h is assumed to be a closed set). We suppose that Ω^h is a quasi-uniform triangulation [5], i.e., there exist positive constants l_1, l_2 and s which are independent of h and such that

$$l_1 h \leq r_i \leq l_2 h\,, \qquad \frac{r_i}{\rho_i} \leq s\,, \qquad i = 1, \ldots, M$$

where r_i and ρ_i are radii of circumscribed and inscribed circles for the triangle τ_i, respectively. We also assume that the triangulation boundary Γ^h approximates Γ with an error $O(h^2)$. If $\Gamma_1 = \Gamma$, we suppose that $\Omega \subset \Omega^h$; if $\Gamma_0 = \Gamma$, we suppose that $\Omega^h \subset \Omega$. If $\Gamma_0 \neq \emptyset$ and $\Gamma_1 \neq \emptyset$, we make the following assumption: points where the boundary condition changes should be at triangulation nodes, $\Gamma_1 \subset \Omega^h$ and $\Gamma_0 \subset \overline{(\mathbb{R}^2 \setminus \Omega^h)}$. Part of Γ^h approximating Γ_0 will be denoted by Γ_0^h, and that for Γ_1 by Γ_1^h. For the triangulation Ω^h, we define the space $H_h(\Omega^h)$ of real continuous functions which are linear on each triangle of Ω^h and vanish at Γ_0^h. We extend these functions on $\Omega \setminus \Omega^h$ by zero.

The solution of the projection problem

$$u^h \in H_h(\Omega^h):\ a(u^h, v^h) = l(v^h) \qquad \forall v^h \in H_h(\Omega^h) \tag{3}$$

will be called an approximate solution of problem (1.2). Aspects of approximation of (1.2) by (1.3) have been thoroughly studied (see [5, 14]); we do not consider them here. Each function $u^h \in H_h(\Omega^h)$ is put in standard correspondence with a real column vector $u \in \mathbb{R}^N$ whose components are values of the function u^h at the corresponding nodes of the triangulation Ω^h. Then

(1.3) is equivalent to the system of mesh equations

$$Au = f$$

$$(Au, v) = a(u^h, v^h) \qquad \forall u^h, v^h \in H_h(\Omega^h) \tag{4}$$

$$(f, v) = l(v^h) \qquad \forall v^h \in H_h(\Omega^h)$$

where u^h and v^h are the respective prolongations of vectors u and v; (f, v) is the Euclidean scalar product in $\mathbb{R}^N$.

The main goal of this work is to construct a symmetric positive definite preconditioning operator B for problem (1.4) so as to satisfy the inequalities

$$c_1(Bu, u) \leq (Au, u) \leq c_2(Bu, u) \qquad \forall u \in \mathbb{R}^N \tag{5}$$

where positive constants c_1 and c_2 are independent of h; the multiplication of a vector by B^{-1} should be easy to implement.

The preconditioner B is constructed by using the method of fictitious space [10] in two stages. At the first stage, we pass from an arbitrary unstructured triangulation Ω^h to an auxiliary structured non-hierarchical mesh, and at the second stage to a hierarchical mesh (a square mesh on a square containing the original domain Ω). Note that the passage from an arbitrary triangulation to a structured mesh was earlier used in [11]. This paper includes some development of [13] for the case of locally refined grids. Another technique for constructing the preconditioners on unstructured meshes was proposed in [8, 9, 10, 17]. The construction of preconditioning operators on non-hierarchical grids was considered in [6].

2 REDUCTION TO A STRUCTURED MESH

The preconditioning operator B in (1.5) is constructed on the basis of the lemma of fictitious space [11]. For convenience, we give this lemma here.

Lemma 2.1. *Let H_0 and H be Hilbert spaces with the scalar products $(u_0, v_0)_{H_0}$ and $(u, v)_H$, respectively. Let A_0 and A be symmetric positive definite continuous operators in the spaces H_0 and H:*

$$A_0\colon\ H_0 \to H_0\,, \qquad A\colon\ H \to H\,.$$

Suppose that R is a linear operator such that

$$R\colon\ H \to H_0$$

$$(A_0 Rv, Rv)_{H_0} \le c_R(Av, v)_H \qquad \forall v \in H$$

and there exists an operator T such that

$$T\colon\ H_0 \to H\,, \qquad RTu_0 = u_0$$

$$c_T(ATu_0, Tu_0)_H \le (A_0 u_0, u_0)_{H_0} \qquad \forall u_0 \in H_0$$

where c_R and c_T are positive constants. Then

$$c_T(A_0^{-1}u_0, u_0)_{H_0} \le (RA^{-1}R^* u_0, u_0)_{H_0} \le c_R(A_0^{-1}u_0, u_0)_{H_0} \qquad \forall u_0 \in H_0\,.$$

The operator R^* is adjoint to R with respect to the scalar products $(u_0, v_0)_{H_0}$ and $(u, v)_H$:

$$R^*\colon\ H \to H_0$$

$$(R^* u_0, v)_H = (u_0, Rv)_{H_0}\,.$$

Note that for constructing and implementing the preconditioner, i.e., the operator $RA^{-1}R^*$, we only require the existence of the operator T. In our case, the role of the operator A_0 is played by A of (1.4), and the role of the space H_0 by $H_h(\Omega_h)$. In order to use Lemma 2.1, we construct a fictitious (auxiliary) space and the corresponding operators. To do this, we embed the domain Ω in a square Π. Let K_i denote the union of triangles in the triangulation Ω^h which have a common vertex z_i, and let d_i be the maximum radius of circle inscribed in K_i. In the square Π, we introduce an auxiliary grid Π_h with a step size $\bar{h}$ such that

$$\bar{h} < \frac{1}{2\sqrt{2}} \min_i\, d_i\,. \tag{6}$$

Let us assume that $\bar{h} = l \cdot 2^{-J}$, where l is the length of sides of Π and J is a positive integer. We denote the nodes of the grid Π^h by Z_{ij},

$$Z_{ij} = (x_i, y_j), \qquad i, j = 0, 1, \dots, L$$

and the cells of Π^h by D_{ij},

$$D_{ij} = \{(x, y) \mid x_i \leq x < x_{i+1},\ y_j \leq y < y_{j+1}\}$$

$$\Pi^h = \bigcup_{i,j=0}^{L} D_{ij}\,.$$

Let Q^h denote the minimum figure that consists of cells D_{ij} and contains Ω^h: $\Omega^h \subset Q^h$; let S^h be the set of boundary nodes of Q^h. We subdivide the set S^h into two subsets S_0^h and S_1^h as follows: if

$$\bar{D}_{ij} \cap \Gamma_0 \neq \oslash$$

all nodes of $D_{ij} \cap S^h$ are in S_0^h

$$S_1^h = S^h \setminus S_0^h\,.$$

Using cell diagonals, we triangulate Q^h and Π^h; hereafter, the designations Q^h and Π^h refer to triangulations as well. Let $H_h(Q^h)$ be the space of real continuous functions which are linear on the triangles of Q^h and vanish at the nodes of S_0^h. It is the space $H_h(Q^h)$ that will be used as the fictitious space in Lemma 2.1.

We now define the projection operator R

$$R\colon\ H_h(Q^h) \to H_h(\Omega^h)$$

the extension operator T

$$T\colon\ H_h(\Omega^h) \to H_h(Q^h)$$

and an easily invertible operator in the space $H_h(\Omega^h)$.

Let us begin with the operator R. For a given mesh function

$$U^h(Z_{ij}) \in H_h(\Omega^h)$$

we define a function $u^h \in H_h(\Omega^h)$ as follows. Let z_l be a vertex in the triangulation Ω^h; assume that $z_l \in D_{ij}$. We put

$$u^h(z_l) = (TU^h)(z_l) = U^h(Z_{ij})\,. \tag{7}$$

The function u^h is equal to zero at nodes $z_l \in \Gamma_0^h$.

Then, let us define the operator T. For a given function $u^h \in H_h(\Omega^h)$, we define a function $U \in H_h(\Omega^h)$. The function U^h is equal to zero at nodes $Z_{ij} \in S_0^h$. At the other nodes, U is defined as follows. If a cell D_{ij} contains a certain vertex z_l of the triangulation Ω^h, we put

$$U^h(Z_{ij}) = (Tu^h)(Z_{ij}) = u^h(z_l)\,.$$

For each of the remaining nodes $Z_{ij} \in Q^h$, we find the closest vertex z_l of the triangulation Ω^h (if there are several closest vertices, we can choose any of them) and put

$$U^h(Z_{ij}) = (Tu^h)(Z_{ij}) = u^h(z_l)\,.$$

Finally, in the space $H_h(Q^h)$ we define the operator A_Q:

$$(A_Q U, V) = \int_{Q_h} ((\nabla U^h, \nabla V^h) + U^h \cdot V^h)\, dx\, dy \qquad \forall U^h, V^h \in H_h(Q^h). \tag{8}$$

where U^h and V^h are the respective prolongations of the vectors U and V.

Theorem 2.1. *There exist positive constants c_3 and c_4, independent of h, such that*

$$c_3(A^{-1}u, u) \le (RA_Q^{-1}R^* u, u) \le c_4(A^{-1}u, u) \qquad \forall u \in \mathbb{R}^N\,.$$

Here A, R and A_Q are operators of (1.4), (2.2) *and* (2.3), *respectively; R^* is the transpose of R (we hereafter use the same designation for an operator and its matrix representation).*

Proof. The theorem easily follows from Lemma 2.1, condition (2.1) and the familiar equivalence of H^1-norms of finite-element functions in the spaces $H_h(\Omega^h)$, $H_h(Q^h)$ and the difference counterparts of these norms [14].

Remark 2.1. The implementation of the operator R is equivalent to the piecewise constant interpolation. It is easily seen that the number of arithmetic operations required for multiplying R or R^* by a vector is proportional to the number of nodes in the mesh domain.

Thus, the construction of a preconditioning operator on an unstructured triangulation is reduced to the construction of a preconditioning operator for A_Q. The latter problem is considered in Section 3.

3 FICTITIOUS SPACE AND MULTILEVEL DECOMPOSITION METHODS

In order to find a preconditioning operator for A_Q, we again use Lemma 2.1. Here the fictitious (auxiliary) space is $H_h(\Pi^h)$ which consists of piecewise linear continuous functions vanishing on the boundary $\partial\Pi$ of the square Π. Efficient preconditioning operators in $H_h(\Pi^h)$ are well known; in particular, we may use the BPX preconditioner [4]. To do so, we use the following construction.

We divide the domain $\Pi \setminus \overline{\Omega}$ into two non-intersecting subdomains such that

$$\begin{aligned} \Pi\setminus = \bar{\Gamma}_0 \cup \bar{\Gamma}_1\,, \qquad & G_0 \cap G_1 = \oslash \\ \partial G_0 \cap \partial\Omega = \Gamma_0\,, \qquad & \partial G_1 \cap \partial\Omega = \bar{\Gamma}_1\,. \end{aligned} \tag{9}$$

According to (3.1), we represent the triangulation $\Pi^h \setminus Q^h$ as a union of two non-overlapping parts:

$$\overline{\Pi^h \setminus Q^h} = G_0^h \cup G_1^h$$

where G_0^h and G_1^h are mesh approximations of the domains G_0 and G_1, respectively. Further, we denote

$$G = \Omega \cup \Gamma_1 \cup G_1\,, \qquad G^h = Q^h \cup G_1^h$$

$H_h(G^h)$ finite-element space of functions vanishing on ∂G^h. We consider in Π^h the sequence of grids

$$\Pi_0^h, \Pi_1^h, \ldots, \Pi_J^h \equiv \Pi^h$$

with step sizes

$$h_0 = l\,, \quad h_1 = l \cdot 2^{-1}\,, \dots\,, h_J \equiv \bar{h} = l \cdot 2^{-J}\,.$$

We triangulate these grids and consider the corresponding finite-element spaces

$$W_0^h \subset W_1^h \subset \dots \subset W_J^h \equiv H_h(\Pi^h)\,.$$

By $\{\Phi_i^{(l)}\}_{i=1}^{N_l}$ we denote the nodal basis of the space W_l^h, $l = 0, 1, \dots, J$.

First, let us examine the case of $\Gamma_1 = \Gamma$; accordingly, here $S_1^h = S^h$. By $\tilde{\Phi}_i^{(l)}$ we denote the restriction of the basic function $\Phi_i^{(l)}$ onto Q^h. We put each function $U^h \in H_h(Q^h)$ in correspondence with a function $\tilde{U}^h \in H_h(\Pi^h)$:

$$\tilde{U}^h(Z_{ij}) = \begin{cases} U^h(Z_{ij})\,, & Z_{ij} \in Q^h \\ 0\,, & Z_{ij} \in \Pi^h \setminus Q^h\,. \end{cases}$$

Define

$$C_N^{-1} U^h = \sum_{l=0}^{J} \sum_{\operatorname{supp} \Phi_i^{(l)} \cap Q^h \neq \emptyset} (\tilde{U}^h, \Phi_i^{(l)})_{L_2(\Pi)}\, \tilde{\Phi}_i^{(l)} \qquad \forall U^h \in H_h(Q^h).$$

Theorem 3.1. *There exist positive constants c_5 and c_6, independent of h, such that*

$$c_5 (A^{-1}u, u) \le (R C_N^{-1} R^* u, u) \le c_6 (A^{-1}u, u) \qquad \forall u \in \mathbb{R}^N\,.$$

Proof. Let us define

$$R_N\colon\ H_h(\Pi^h) \to H_h(Q^h)$$

to be an operator of restriction on Q^h:

$$(R_N U^h)(Z_{ij}) = U^h(Z_{ij}) \qquad \forall Z_{ij} \in Q^h\,.$$

If we subdivide the nodes of Π^h into two groups: (1) the nodes of Q^h (including those of S^h), and (2) the remaining nodes, then we obtain the following matrix representation for R_N (see also [1]):

$$R_N = (IO)$$

where I is the identity matrix corresponding to nodes of group (1), and O is the zero matrix corresponding to nodes of group (2). It is evident that

$$\|R_N U^h\|_{H^1(Q^h)} \le \|U^h\|_{H^1(\Pi^h)} \qquad \forall U^h \in H_h(\Pi^h).$$

By the theorem of extension of mesh functions [6], there exists the extension operator

$$T_N\colon\ H_h(Q^h) \to H_h(\Pi^h)$$

uniformly bounded with respect to h.

According to Lemma 2.1 and [4], there exist positive constants c_7 and c_8, independent of h, such that

$$c_7(A_Q^{-1}U, U) \le (R_N C_\Pi^{-1} R_N^* U, U) \le c_8(A_Q^{-1}u, u) \qquad \forall U$$

where A_Q is the operator of (2.3) and the definition of C_Π^{-1} is

$$C_\Pi^{-1}U^h = \sum_{l=0}^{J} \sum_{i=1}^{N_l} (U^h, \Phi_i^{(l)})_{L_2(\Pi)}\, \Phi_i^{(l)} \qquad \forall U^h \in H_h(\Pi^h).$$

Taking into account the explicit form of R_N, we complete the proof of Theorem 3.1.

Then, let us examine the case of the Dirichlet problem, i.e., $\Gamma_0 = \Gamma$ and, accordingly, $S_0^h = S^h$. We define the preconditioner as follows:

$$C_D^{-1}U^h = \sum_{l=0}^{J} \sum_{\operatorname{supp}\Phi_i^{(l)} \subset Q^h} (U^h, \Phi_i^{(l)})_{L_2(Q^h)}\, \Phi_i^{(l)} \qquad \forall U^h \in H_h(Q^h).$$

Theorem 3.2. *There exist positive constants c_9 and c_{10}, independent of h, such that*

$$c_9(A^{-1}u, u) \le (RC_D^{-1}R^*u, u) \le c_{10}(A^{-1}u, u) \qquad \forall u \in \mathbb{R}^N.$$

Proof. In this case, the equivalence of the operators A_Q and C_D easily follows from the multilevel technique [3, 4, 15, 16] and can be done, for instance, by using quasi-interpolants from [12]. Then, from Theorem 2.1 we get the assertion of Theorem 3.2.

Finally, we examine the case of mixed boundary conditions, i.e., $\Gamma_0 \neq \oslash$ and $\Gamma_1 \neq \oslash$. We denote

$$C_M^{-1} U^h = \sum_{l=0}^{J} \sum_{\substack{\operatorname{supp} \Phi_i^{(l)} \subset G^h, \\ \operatorname{supp} \Phi_i^{(l)} \cap Q^h \neq \oslash}} (\tilde{U}^h, \Phi_i^{(l)})\, \tilde{\Phi}_i^{(l)} \qquad \forall U^h \in H_h(Q^h).$$

Theorem 3.3. *There exist positive constants c_{11} and c_{12}, independent of h, such that*

$$c_{11}(A^{-1}u, u) \leq (RC_M^{-1}R^* u, u) \leq c_{12}(A^{-1}u, u) \qquad \forall u \in \mathbb{R}^N .$$

Proof. The theorem is proved by using the argument of Theorem 3.2 and then that of Theorem 3.1. Indeed, at the first step, let us 'extend' the Dirichlet boundary condition from S_0^h to the boundary of the triangulation Π^h. To do it, we consider finite element space $H_h(G^h)$ and define

$$C_G^{-1} U^h = \sum_{l=0}^{J} \sum_{\operatorname{supp} \Phi_i^{(l)} \subset G^h} (U^h, \Phi_i^{(l)})_{L_2(G^h)}\, \Phi_i^{(l)} \qquad \forall U^h \in H_h(G^h).$$

Then, according to Theorem 3.2, there exist positive constants c_{13}, c_{14}, independent of h, such that

$$c_{13}\|U^h\|^2_{H^1(G^h)} \leq (C_G U, U) \leq c_{14}\|U^h\|^2_{H^1(G^h)} \qquad \forall U^h \in H_h(G^h).$$

At the second step, define

$$R_{N,G}\colon\ H_h(G^h) \to H_h(Q^h)$$

as a restriction on Q^h from G^h:

$$(R_{N,G}U^h)(Z_{ij}) = U^h(Z_{ij}) \qquad \forall Z_{ij} \in Q^h .$$

Then, from Lemma 2.1 we get

$$c_{15}(A_Q^{-1}U, U) \leq (R_{N,G}C_G^{-1}R_{N,G}^* U, U) \leq c_{16}(A_Q^{-1}U, U) \qquad \forall U^h \in H_h(Q^h)$$

where c_{15}, c_{16} are independent of h. Using again the explicit form of $R_{N,G}$, we complete the proof of Theorem 3.3.

4 LOCALLY REFINED GRIDS

In this section we consider a triangulation Ω^h of the domain Ω

$$\Omega^h = \bigcup_{i=1}^{M} \tau_i$$

and assume Ω^h is regular but not quasi-uniform, i.e., there exists a constant s, independent of h, such that

$$\frac{r_i}{\rho_i} \leq s\,, \qquad i = 1, \ldots, M$$

where r_i and ρ_i are radii of circumscribed and inscribed circles for the triangle τ_i, respectively. It means that Ω^h can be locally refined. For this triangulation Ω^h, we define the space $H_h(\Omega^h)$ of real continuous functions which are linear on each triangle τ_i of Ω^h. For the sake of simplicity, we consider the Dirichlet boundary condition and assume that the functions from $H_h(\Omega^h)$ vanish at Γ^h.

If we introduce a uniform fictitious grid Q^h, then it is possible to modify the operators R and T from Section 2 for locally refined triangulation Ω^h, but realization of a preconditioner will be expensive.

Let us embed the domain Ω in a square Π and start with a coarse uniform grid Π_0^h. We refine Π_0^h several times

$$\Pi_0^h, \Pi_1^h, \ldots$$

The grid Π_l^h consists of cells $D_{ij}^{(l)}$. Let Q_0^h denote the minimum figure that consists of cells $D_{ij}^{(0)}$ and contains Ω^h. Denote by I_0 a set of indices (i,j) such that

$$Q_0^h = \bigcup_{(i,j)\in I_0} D_{ij}^{(0)}$$

We define grids $Q_1^h, Q_2^h, \ldots$ in the following way. Denote by I_l a set of indices (i,j) such that the cell $D_{ij}^{(l)}$ contains more than one vertex of the triangulation Ω^h. We divide $D_{ij}^{(l)}$ and all neighboring cells (which have at least one common node with $D_{ij}^{(l)}$) into four congruent sub cells by connecting the midpoints of the edges. Denote new cells by $D_{ij}^{(l+1)}$ and a resulting grid by $Q_{l+1}^h, l = 0, 1, \ldots$, which are the minimum figure that contains Ω^h. We stop this process when each cell contains no more than one vertex of Ω^h. Denote by Q_J^h the final grid.

Define a finite-element space $H_h(Q^h)$ as follows:

$$H_h(Q^h) = \{ \sum_{\operatorname{supp}\Phi_k^{(0)} \subset Q_J^h} \alpha_k^{(0)} \Phi_k^{(0)} + \sum_{l=0}^{J-1} \sum_{(i,j)\in I_l} \sum_{\operatorname{supp}\Phi_k^{(l+1)} \cap D_{ij}^{(l)} \neq \emptyset} \alpha_k^{(l+1)} \Phi_k^{(l+1)} \mid \alpha_k^{(l)} \in \mathbb{R}, \}$$

We now define the projection operator R

$$R\colon\ H_h(Q^h) \to H_h(\Omega^h)$$

the extension operator T

$$T\colon\ H_h(\Omega^h) \to H_h(Q^h)$$

according to the definitions from Section 2.

Define a preconditioning operator in $H_h(Q_J^h)$ in the following way:

$$C_R^{-1} U^h = \sum_{\operatorname{supp}\Phi_k^{(0)} \subset Q_J^h} (U^h, \phi_k^{(0)})_{L_2(Q_J^h)} \Phi_k^{(0)} + \sum_{l=0}^{J-1} \sum_{(i,j)\in I_l} \sum_{\operatorname{supp}\Phi_k^{(l+1)} \cap D_{ij}^{(l)} \neq \emptyset} (U^h, \phi_k^{(l+1)})_{L_2(Q_J^h)} \Phi_k^{(l+1)}$$

for any $U^h \in H_h(Q_J^h)$.

Theorem 4.1 *There exist positive constants c_{17} and c_{18}, independent of h, such that*

$$c_{17}(A^{-1}u, u) \leq (RC_R^{-1}R^* u, u) \leq c_{18}(A^{-1}u, u) \qquad \forall u \in \mathbb{R}^N .$$

Proof. In this case, we again use the equivalence of H^1-norms of finite-element functions in the spaces $H_h(\Omega^h)$, $H_h(Q^h)$ and the difference counterparts of these norms and the multilevel technique.

References

[1] G. P. Astrakhantsev, Fictitious domain method for the second-order elliptic equation with natural boundary conditions. *Zh. Vychisl. Mat. Mat. Fiz.*, **18** (1978), pp. 118–125.

[2] J.-P. Aubin, *Approximation of Elliptic Boundary-Value Problems.* Wiley–Interscience, New York–London–Sydney–Toronto, 1972.

[3] F. A. Bornemann and H. Yserentant, A basic norm equivalence for the theory of multilevel methods. *Numer. Math.*, **64** (1993), pp. 455–476.

[4] J. H. Bramble, J. E. Pasciak and J. Xu, Parallel multilevel preconditioners, *Math. Comp.*, **55** (1990), pp. 1–22.

[5] Ph. Ciarlet, *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1977.

[6] R. Kornhuber and H. Yserentant, Multilevel methods for elliptic problems on domains not resolved by the coarse grid. *Domain decomposition for PDEs, D.E. Keyes and J. Xu eds., Contemporary Mathematics*, **180** (1994), pp. 49–60.

[7] A. M. Matsokin, Extension of mesh functions with norm-preserving. In: *Variational Methods of Numerical Analysis*, Comp. Centre, Siberian Branch of Acad. Sci. of the USSR, Novosibirsk, 1986, pp. 111–132 (in Russian).

[8] A. M. Matsokin, Solution of grid equations on non-regular grids. *Preprint No. 738*, Comp. Centre, Siberian Branch of Acad. Sci. of USSR, Novosibirsk, 1987 (in Russian).

[9] A. M. Matsokin and S. V. Nepomnyaschikh, The fictitious domain method and explicit continuation operators. *Zh. Vychisl. Mat. Mat. Fiz.*, **33** (1993), pp. 45–59.

[10] S. V. Nepomnyaschikh, Method of splitting into subspaces for solving elliptic boundary value problems in complex-form domains. *Sov. J. Numer. Anal. Math. Model.*, **6** (1991), No. 2, pp. 151–168.

[11] S. V. Nepomnyaschikh, Mesh theorems of traces, normalization of function traces and their inversion. *Sov. J. Numer. Anal. Math. Model.*, **6** (1991), No. 3, pp. 223 – 242.

[12] S. V. Nepomnyaschikh, Optimal multilevel extension operators. *Preprint SPC 95-3, Technische Universitate Chemnitz-Zwickau*, 1995.

[13] S. V. Nepomnyaschikh, Fictitious space method on unstructured meshes. *East-West J. Numer. Math.*, **3** (1995), No. 1 (to appear).

[14] L. A. Oganesyan and L. A. Rukhovets, *Variational Difference Methods for Solving Elliptic Equations.* Izdat. Akad. Nauk Arm. SSR, Erevan, 1979 (in Russian).

[15] P. Oswald, *Multilevel Finite Element Approximation: Theory and Application.* B. G. Teubner, Stuttgart, 1994.

[16] J. Xu, Iterative methods by space decomposition and subspace correction. *SIAM Review*, **34** (1992), No. 4, pp. 581 – 613.

[17] J. Xu, The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids (submitted to *Computing*).

[18] G. N. Yakovlev, On traces of piecewise smooth surfaces of functions from the space W_p^l. *Mat. Sbornik* **74**, (1967), pp. 526 – 543.

Page intentionally left blank

MULTIGRID METHODS FOR EHL PROBLEMS

Elyas Nurgat and Martin Berzins
School of Computer Studies, University of Leeds
Leeds, LS2 9JT, UK

INTRODUCTION

In many bearings and contacts, forces are transmitted through thin continuous fluid films which separate two contacting elements. Objects in contact are normally subjected to friction and wear which can be reduced effectively by using lubricants. If the lubricant film is sufficiently thin to prevent the opposing solids from coming into contact and carries the entire load, then we have hydrodynamic lubrication, where the lubricant film is determined by the motion and geometry of the solids. However, for loaded contacts of low geometrical conformity, such as gears, rolling contact bearings and cams, this is not the case due to high pressures and this is referred to as Elasto-Hydrodynamic Lubrication (EHL) (ref. 1). In EHL, elastic deformation of the contacting elements and the increase in fluid viscosity with pressure are very significant and cannot be ignored.

Since the deformation results in changing the geometry of the lubricating film, which in turn determines the pressure distribution, an EHL mathematical model must simultaneously satisfy the complex elasticity (integral) and the Reynolds lubrication (differential) equations. The nonlinear and coupled nature of the two equations makes numerical calculations computationally intensive. This is especially true for highly loaded problems found in practice. One novel feature of these problems is that the solution may exhibit sharp pressure spikes in the outlet region (ref. 1).

To this date both finite element and finite difference methods have been used to solve EHL problems with perhaps greater emphasis on the use of the finite difference approach. In both cases, a major computational difficulty is ensuring convergence of the nonlinear equations solver to a steady state solution. Two successful methods for achieving this are direct iteration and multigrid methods.

Direct iteration methods (e.g Gauss Seidel) have long been used (e.g Hamrock and Dowson (ref. 2)) in conjunction with finite difference discretizations on regular meshes. Perhaps one of the best examples of the application of such methods is the recent Effective Influence Method of Dowson and Wang (ref. 3). Multigrid methods have also been used with great success by Venner (ref. 4) and Venner and Lubrecht (ref. 5) with a good summary being given by Venner (ref. 6).

As both these finite difference discretization based approaches appear to provide an efficient way of solving EHL problems, it is important to understand their relative merits. This paper is a first attempt at providing such an understanding in the context of EHL point contact problem, (contact of two spheres), in which the contact zone is a point and an ellipse or circle for unloaded and loaded dry contacts respectively. Since the film thickness and the contact width are generally small compared to the local radius of curvature of the two surfaces, the reduced geometry of the surfaces in the contact area can be accurately approximated to the contact between a paraboloid and a flat surface.

The layout of the remainder of this paper is as follows. In section 2 we introduce the form of the equations to be solved. The Effective Influence Newton Method is described in Section 3 while Section 4 describes the Multigrid method to be used. Sections 5 and 6 describe the test problems to be used in the comparison between the two methods and compare the performance of the two methods. Section 7 concludes the paper with an argument of the two methods and suggests some future research directions.

GOVERNING EQUATIONS

The Mathematical model describing the isothermal axisymmetric EHL circular contact problem consists of three equations. The Reynolds Equation relates pressure, P, to geometry of the gap, the film thickness, H, and velocities of the running surfaces.

$$L(P) = \frac{\partial}{\partial x}\left(\epsilon\frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial y}\left(\epsilon\frac{\partial P}{\partial y}\right) - \frac{\partial(\rho H)}{\partial x} = 0\,,\ x,y \in [-3.5, 1.5] \times [-2, 2] \tag{1}$$

with the cavitation condition $P \geq 0$ and $P = 0$ on boundaries. The function $\epsilon = (\rho H^3)/(\eta\lambda)$ depends on viscosity, $\eta(P)$, density, $\rho(P)$, and film thickness, $H(x,y)$. The remaining terms are given by:

$\rho = \begin{cases} 1 + \frac{\mu p_h P}{1+v p_h P} & \text{if } P > 0 \\ 1 & \text{otherwise} \end{cases}$, (ref. 6);

$\eta = \exp\{\frac{\alpha p_0}{z}[-1 + (1 + \frac{p_h}{p_0}P)^z]\}$, (ref. 6);

p_h is the maximum Hertzian pressure given by $p_h = \frac{L}{\alpha\pi}\sqrt[3]{\frac{3M}{2}}$;

$\alpha =$ pressure viscosity coefficient, $z = 0.68$ is the pressure viscosity index;

$\lambda = \frac{4\pi}{M}\sqrt[3]{\frac{2}{3M}}$ and $p_0 = 1.98 \times 10^8$ are constants;

$\mu = 5.8 \times 10^{-10}$ and $v = 1.68 \times 10^{-9}$ are empirical constants;

L and M are the Moes (ref. 6) dimensionless material and load parameters, respectively. For lightly loaded problems p_h, which is a function of M and L, is about 0.5 GPa. Moderately loaded problems have p_h in the range of about 1 GPa.

The Film Thickness Equation, $H(x,y)$, computes the elastic distortion of the surfaces caused by the pressure in the film and is written as:

$$H(x,y) = H_{00} + \frac{x^2}{2} + \frac{y^2}{2} + \frac{2}{\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{P(x',y')\,dx'\,dy'}{\sqrt{(x-x')^2 + (y-y')^2}} \tag{2}$$

where H_{00} is a constant.

The final equation is the Force Balance Equation which ensures that the integral over the pressure balances the external applied load:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x,y)\,dx\,dy = ExternalForce. \tag{3}$$

The nondimensionalisation employed allows the external force to be scaled to $(2\pi)/3$.

Finite Difference Discretization of Governing Equations

The focus of this study is on the iterative solution methods for the nonlinear equations and so in order to allow comparison with existing results we shall follow most EHL studies and use a regular mesh. The governing equations are discretized on a regular rectangular grid with the direction of flow in the x direction and the mesh spacings h_x and h_y in the x and y directions, respectively. Due to symmetry, only half the domain is used in the y direction. Reynolds Equation (1) is discretized at each non boundary mesh point (i,j), $((i-1)h_x + x_a, (j-1)h_y - y_c)$ where $x, y \in [x_a, x_b] \times [-y_c, y_c]$, using central and backward differencing to get, (ref. 6),

$$\epsilon_{i-\frac{1}{2},j}(P_{i-1,j} - P_{i,j}) + \epsilon_{i+\frac{1}{2},j}(P_{i+1,j} - P_{i,j}) + h_x^2 h_y^{-2}(\epsilon_{i,j-\frac{1}{2}}(P_{i,j-1} - P_{i,j}) + \epsilon_{i,j+\frac{1}{2}}(P_{i,j+1} - P_{i,j})) - h_x(\rho_{i,j}H_{i,j} - \rho_{i-1,j}H_{i-1,j}) = 0 \tag{4}$$

where $\epsilon_{i+\frac{1}{2},j}, \epsilon_{i-\frac{1}{2},j}, \epsilon_{i,j+\frac{1}{2}}, \epsilon_{i,j-\frac{1}{2}}$ denote the values of ϵ at the intermediate locations midway between meshpoints.

The discretized film thickness equation (2) at a point (i,j) is given by:

$$H_{i,j} = H_{00} + \frac{x_i^2}{2} + \frac{y_j^2}{2} + d_{i,j} \tag{5}$$

where H_{00} is a constant and $d_{i,j}$ is the elastic deformation of the material due to the applied load as defined below.

Elastic Deformation Integral

The elastic deformation on the surface of a solid depends on the representation of applied normal pressures. The simplest procedure is to divide the pressure distribution into rectangular blocks of uniform pressure. The elastic deformation at a point (x,y), $d_{x,y}$, due to the uniform pressure over the rectangular area $2a2b$ is given by (ref. 6) :

$$d_{x,y} = \frac{2P}{\pi^2} \int_{-b}^{b} \int_{-a}^{a} \frac{dx_1\, dy_1}{\sqrt{(x-x_1)^2 + (y-y_1)^2}} \,. \tag{6}$$

If the entire domain is divided into equal rectangular areas, then from Dowson and Hamrock (ref. 7), the elastic deformation at a point (i,j), $d_{i,j}$, due to contributions of all rectangular areas of uniform pressure is given by:

$$d_{i,j} = \frac{2}{\pi^2} \sum_{k=1}^{m_x} \sum_{l=1}^{n_y} K_{m,n} P_{k,l} \tag{7}$$

where $m = |i-k| + 1$, $n = |j-l| + 1$, m_x and n_y are the maximum number of points in the x and y directions, respectively. The coefficients $K_{m,n}$ are given by:

$$|x_p| \ln\left(\frac{y_p + \sqrt{x_p^2 + y_p^2}}{y_q + \sqrt{x_p^2 + y_q^2}}\right) + |y_q| \ln\left(\frac{x_q + \sqrt{y_q^2 + x_q^2}}{x_p + \sqrt{y_q^2 + x_p^2}}\right) + |x_q| \ln\left(\frac{y_q + \sqrt{x_q^2 + y_q^2}}{y_p + \sqrt{x_q^2 + y_p^2}}\right) + |y_p| \ln\left(\frac{x_p + \sqrt{y_p^2 + x_p^2}}{x_q + \sqrt{y_p^2 + x_q^2}}\right)$$

where

$$x_p = x_i - x_k + \tfrac{h_x}{2} \qquad x_q = x_i - x_k - \tfrac{h_x}{2} \qquad y_p = y_j - y_l + \tfrac{h_y}{2} \qquad y_q = y_j - y_l - \tfrac{h_y}{2} \ .$$

One advantage of a regular mesh is that the $m_x n_y$ coefficients need only be calculated once and stored. In contrast, on an irregular mesh it is necessary to store $m_x n_y$ coefficients for each mesh point.

The force balance equation (3) determines the value of the integration constant H_{00} and is discretized as follows:

$$h_x h_y \sum_{i=1}^{m_x} \sum_{j=1}^{n_y} P_{i,j} - \frac{2\pi}{3} = 0 \ . \qquad (8)$$

The system of equations (4), (7) and (8) thus constitutes a system of integro-differential equations. The initial pressure distribution is given by the Hertzian pressure profile, (ref. 6). That is $P = \sqrt{1 - x^2 - y^2}$ if $x^2 + y^2 < 1$ otherwise $P = 0$.

EFFECTIVE INFLUENCE NEWTON METHOD, [ref. 8]

For EHL problems, when Newton's method is used, the discretized nonlinear equation is linearized and solved using Gaussian elimination or an iteration method. Gaussian elimination may be used if the dimension of the coefficient matrix, Jacobian matrix, of the linear system is small. For EHL problems, a full Jacobian matrix is required because the elastic deformation at one point is determined by the pressure distribution over the entire grid. For a mesh of m_x, n_y points, this results in an often prohibitively large dense system of $m_x n_y$ equations. It is thus essential to seek computationally less expensive methods.

The Effective Influence Newton Method developed by Wang (ref. 8), to solve EHL problems, is a variant of Newton's method for solving nonlinear equations. This method employs the notion of effective influence to determine the contribution from elastic deformation in the solution of the set of approximate linear equations used in Newton's formulation of the EHL problem. The elastic deformation at a point (i, j) is and must be determined by the pressure distribution over the entire domain, though the contribution decreases radially outwards. However, when obtaining the solution of the linearized Reynolds equation by Newton's method, pressures not close to the point (i, j) can be ignored.

The elastic deformation at a point (i, j) due to a rectangular area of uniform pressure at some other point is strongly influenced by the distance between the two points. This enables us to define an effective influence region such that only the pressures within the region need to be considered when solving the approximate linearized Reynolds equation. This results in a banded, rather than full, Jacobian matrix, thus reducing the computational work involved in the EHL calculation.

Suppose $\underline{P}$ is an approximation to the true solution $\underline{\tilde{P}}$, then at a point (i, j), $L_{i,j} = L(\underline{P})_{i,j} \neq 0$ and $\tilde{L}_{i,j} = L(\underline{\tilde{P}})_{i,j} = 0$. Taylor's theorem gives:

$$\tilde{L}_{i,j} = L_{i,j} + \sum_{l=1}^{n_y} \sum_{k=1}^{m_x} \frac{\partial L_{i,j}}{\partial P_{k,l}} \Delta P_{k,l} + O((\Delta P)^2) \qquad (9)$$

where $L_{i,j}$ is the discretized Reynolds equation (1) at the point (x_i, y_j).

If (m_i) and (n_j) are the number of effective points, from the point (i,j), in the x and y directions, respectively, then the Effective Influence Newton's formula is of the form:

$$\sum_{l=j-n_j}^{j+n_j} \sum_{k=i-m_i}^{i+m_i} \frac{\partial L_{i,j}}{\partial P_{k,l}} \Delta P_{k,l} + L_{i,j} = 0 \; . \tag{10}$$

The simplest form of the Effective Influence Newton's method makes use of five adjacent nodal points in linearizing the original Reynolds Equation. This is the method employed by Dowson and Wang (ref. 3) in solving the EHL problems and is of the form:

$$\frac{\partial L_{i,j}}{\partial P_{i-1,j}} \Delta P_{i-1,j} + \frac{\partial L_{i,j}}{\partial P_{i,j}} \Delta P_{i,j} + \frac{\partial L_{i,j}}{\partial P_{i+1,j}} \Delta P_{i+1,j} = -L_{i,j} - \frac{\partial L_{i,j}}{\partial P_{i,j-1}} \Delta P^{new}_{i,j-1} - \frac{\partial L_{i,j}}{\partial P_{i,j+1}} \Delta P^{old}_{i,j+1} \; . \tag{11}$$

For a constant j, equation (11) results in a tridiagonal system of equations which are solved simultaneously using I-line relaxation, provided that $\Delta P^{new}_{i,j-1}$ and $\Delta P^{old}_{i,j+1}$ are known. In every iteration the correction term $\Delta P_{i,j}$ is evaluated on the entire grid. Having obtained $\Delta \underline{P}$, a new approximation $\overline{P}_{i,j}$ to $P_{i,j}$ is computed on the entire grid using:

$$\overline{P}_{i,j} = P_{i,j} - W \Delta P_{i,j} \tag{12}$$

where W is a damping factor in the range 0.09 to 0.2.

The new values of pressure are then used to calculate the elastic deformation, $d_{i,j}$, and the film thickness constant, H_{00}, of the film thickness equation (5). H_{00} is updated using the force balance equation (8) and is given by:

$$H_{00} = H_{00} - c(\frac{2\pi}{3} - h_x h_y \sum_{i=1}^{m_x} \sum_{j=1}^{n_y} P_{i,j}) \tag{13}$$

where c is a small constant taken, here as 10^{-2}.

The technique employed to analyze the convergence of the solution is based on the change in the solution from one iteration to the next. Thus, the ERROR on the k^{th} iteration is given by:

$$ERROR = \frac{\sum_{i=1}^{m_x} \sum_{j=1}^{n_y} |P^k_{i,j} - P^{k-1}_{i,j}|}{\sum_{i=1}^{m_x} \sum_{j=1}^{n_y} P^k_{i,j}} \tag{14}$$

and the iteration is terminated when ERROR < TOL, where TOL is a user supplied tolerance. The results of Dowson and Wang (ref. 3) and (ref. 8) show that the method works well for many different types of EHL problems.

MULTIGRID METHOD

The use of multigrid methods in solving EHL problems is relatively new. This method was introduced into the field of Tribology by Lubrecht (ref. 9), who through his extensive work has made multigrid techniques an important technique for solving EHL problems. The use of multigrids for solving EHL line and point contact problems has been described by Venner (ref. 6).

The concept of multigrid iteration depends on the asymptotic nature of errors associated with iterative schemes and how the schemes reduce these errors. Smooth error components associated with low frequencies are hardly reduced with the classical iterative schemes, thus resulting in a long time to converge. The opposite is true for error components with wavelength of the order of the meshsize. However, low frequency error components can be adequately represented on coarser grids. In a multilevel solver, which makes use of a series of coarser grids, each error component is solved until the component becomes smooth when the procedure is switched onto a coarser grid.

Full Approximation Scheme

FDMG Multigrid Software of Gareth Shaw (ref. 10) is used as a starting point for implementing the multigrid technique. FDMG employs Multigrid Full Approximation Scheme (FAS) to solve nonlinear systems of partial differential equations using either V or W coarse grid correction cycle. Jacobi or Gauss-Seidel iterative method can be used as a smoother. The option for the type of restriction is either injection or full weighting.

EHL problems are nonlinear, thus when using multigrids the standard Correction Scheme can not be used; instead the Full Approximation Scheme must be used. In the cavitation region, in which negative pressures are computed by the solver, the Reynolds equation is not valid and the computed pressures are set to zero in the standard manner (ref. 6). This is treated with the multigrid method by using injection near and in the cavitational region when transferring the residual and solution to the coarse grid. Full weighting is used in the remaining part of the domain. The elastic deformation and the force balance equation gets updated on each grid using the updated pressure values. The only substantial modification to FDMG has been to take symmetry boundary conditions and cavitation into consideration. The main difference from the scheme of Venner (ref. 6) is that he uses a combination of Jacobi and Gauss Seidel rather than the Gauss Seidel scheme used here.

Relaxation

The solution for the isothermal point contact problem is obtained by I-line relaxation due to strong coupling in the direction of flow, x direction. The discrete equations are solved simultaneously on a line of points, sweeping across the grid only in the positive y direction due to symmetry. On each line of points, the Effective Influence Method is employed, as described above, and a tridiagonal system of equations is solved. The criterion for convergence are based on comparing the solutions on two grids with meshsize h and $H = 2h$. Thus the error, ERR(h,H), as used by Venner (ref. 6) to measure convergence is given by:

$$ERR(h,H) = h_x h_y \sum_{i=1}^{m_x} \sum_{j=1}^{n_y} |\tilde{p}_{i,j}^{H} - I_h^H \tilde{p}_{i,j}^{h}| \ . \tag{15}$$

TEST PROBLEM ONE

This test problem, which appears in Wang (ref. 8), is solved on a single 151 by 81 grid of domain $\{(x, y) : -3.5 \leq x \leq 1.5, -2.0 \leq y \leq 2.0\}$. For this moderately loaded problem, the values of Moes (ref. 6) dimensionless parameters are $M = 99$ and $L = 16$. This in turn gives $\lambda = 2.397494 \times 10^{-2}$. The maximum Hertzian pressure, p_h, at this load is 1.21 GPa if $\alpha = 2.205645 \times 10^{-8}$. The equivalent Hamrock and Dowson's (ref. 11) dimensionless parameters with U fixed at 5.6102×10^{-11} are $W = 3.4125 \times 10^{-6}$ and $G = 4865$.

This problem is solved by using the Effective Influence Newton method for 1500 iterations. Every 50 iterations the minimum, Hmin, and central, Hcent, film thickness is recorded. Table 1 shows Hcent and Hmin together with the equivalent minimum film thickness of Hamrock and Dowson, HDHmin. The minimum and central film thickness achieved by Wang (ref. 8) after 100 iterations is 0.28827×10^{-4} at (I,J)=(113,24). Figure 1 shows the profiles of the pressure and film thickness along the x-axis. The pressure spike near the outlet is an often observed feature of EHL solutions.

Its	Hcent	Hmin @ (I,J)	HDHmin	RMSRES	SumP	ERROR
50	0.2679E+00	0.1170E+00 (126, 1)	0.3476E-04	0.171E-02	0.9529	0.144E-01
100	0.1316E+00	0.5548E-01 (113,18)	0.1648E-04	0.158E-02	1.7756	0.616E-02
150	0.1505E+00	0.7472E-01 (111,19)	0.2219E-04	0.149E-02	2.0660	0.106E-02
200	0.1683E+00	0.8592E-01 (111,19)	0.2552E-04	0.143E-02	2.1125	0.294E-03
250	0.1787E+00	0.9148E-01 (111,19)	0.2717E-04	0.137E-02	2.1109	0.234E-03
300	0.1849E+00	0.9366E-01 (114,18)	0.2782E-04	0.131E-02	2.1052	0.171E-03
350	0.1891E+00	0.9504E-01 (114,18)	0.2823E-04	0.126E-02	2.1017	0.123E-03
400	0.1922E+00	0.9610E-01 (114,18)	0.2854E-04	0.122E-02	2.0996	0.931E-04
450	0.1946E+00	0.9689E-01 (113,18)	0.2878E-04	0.117E-02	2.0984	0.739E-04
500	0.1965E+00	0.9753E-01 (113,18)	0.2897E-04	0.113E-02	2.0977	0.605E-04
750	0.2024E+00	0.9949E-01 (113,18)	0.2955E-04	0.947E-03	2.0958	0.283E-04
1000	0.2053E+00	0.1004E+00 (113,18)	0.2983E-04	0.795E-03	2.0952	0.163E-04
1500	0.2082E+00	0.1013E+00 (113,18)	0.3008E-04	0.583E-03	2.0947	0.702E-05

Table 1: Test Problem One on a single 151 by 81 grid, M=99 & L=16

Convergence Criteria. Table 1 also shows the errors, associated with the solution, from which the accuracy of the solution can be analyzed. If the convergence criteria are based, as in Wang, see equation (14), (ref. 8), on the change in the solution from one iteration to the next, labelled ERROR in Table 1, then the solution has converged to the order of 10^{-5}. After 100 iterations the solution has converged to the order of 10^{-2} and the corresponding error value found by Wang (ref. 8) is 0.182×10^{-3} on the same grid.

The sum of the pressures over the entire grid, labelled SumP in Table 1, also suggests that the iteration is converging as the sum of pressures on the final iteration is converging towards 2.0943, thus obeying the force balance equation (8).

However, if the convergence is based on the Root Mean Square Residual, labelled RMSRES in Table 1, then it can be said that the solution may not have completely converged. The reason for this is due to the nature of the Reynolds equation. The coefficient ϵ of the Reynolds equation plays a vital role in the solving of these equations. The pressures in the contact region, $x^2 + y^2 < 1$, are larger than those in the non contact region. This makes the coefficient ϵ vary by several orders of magnitude over the computational domain. Consider the case along the line of symmetry, y=0. In the contact region ϵ is very small ranging from 10^{-9} to 10^{-2}, whereas in the non contact region ϵ varies from 10^{-1} to 10^{4} as can be seen from Figure 2. Thus, when ϵ is very small the film thickness derivative part of the Reynolds equation dominates, whereas when ϵ is large the contribution from the film thickness derivative part is minimal. Figure 2 also shows that the residuals are between two and four orders of magnitude smaller in the contact region than in the inlet and outlet regions.

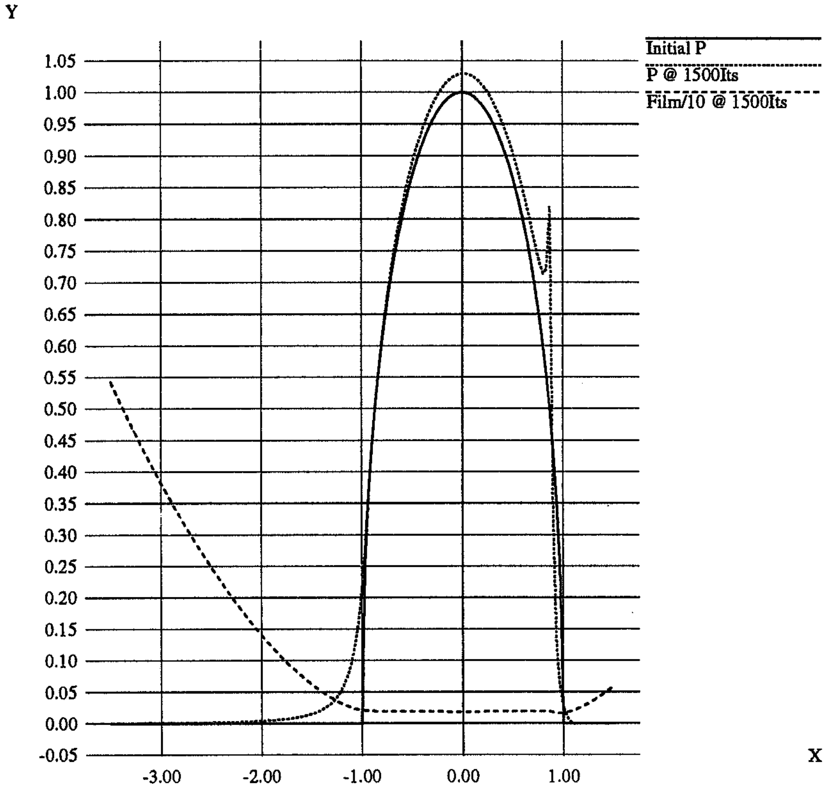


Figure 1: Pressure and Film profiles along y=0, Test Problem One.

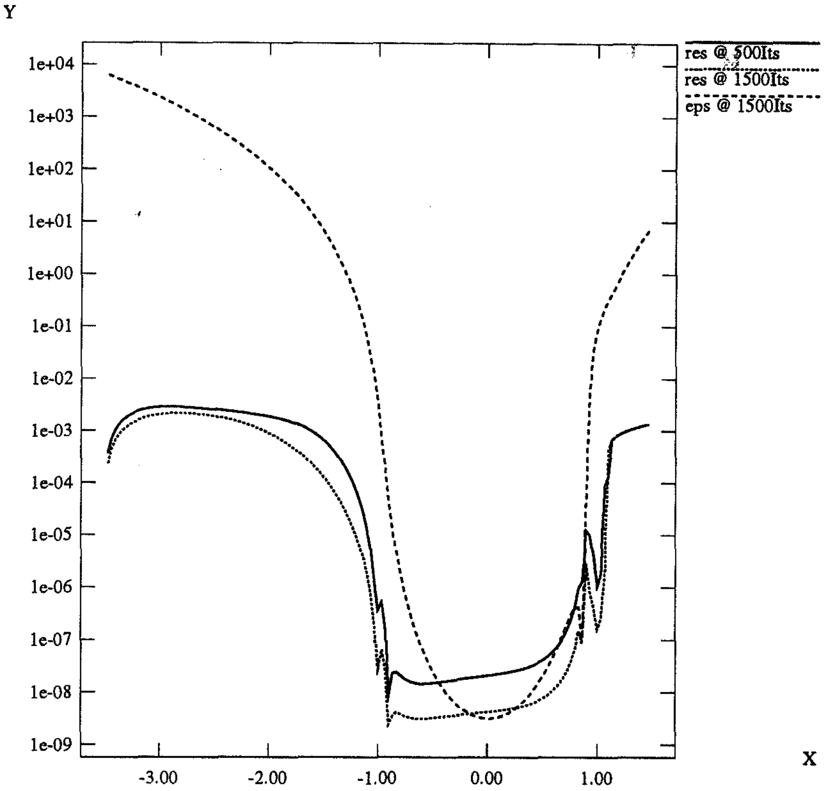


Figure 2: Residual and Eps, ϵ, profiles along y=0, Test Problem One.

Its	Hcent	Hmin @ (I,J)	RMSRES	SumP	ERR(4,3)
10	0.187E+00	0.140E+00 (80, 1)	0.34945E-02	1.6123	0.8557E-02
20	0.109E+00	0.618E-01 (94,31)	0.32213E-02	2.1197	0.1654E-02
30	0.143E+00	0.772E-01 (95,30)	0.31004E-02	2.1049	0.1021E-02
40	0.157E+00	0.831E-01 (95,30)	0.30013E-02	2.0965	0.6970E-03
50	0.166E+00	0.864E-01 (96,29)	0.29144E-02	2.0924	0.5218E-03
60	0.172E+00	0.887E-01 (96,29)	0.28358E-02	2.0907	0.4102E-03
70	0.177E+00	0.907E-01 (96,29)	0.27631E-02	2.0889	0.3278E-03
80	0.181E+00	0.922E-01 (96,29)	0.26951E-02	2.0874	0.2913E-03
90	0.184E+00	0.934E-01 (96,29)	0.26307E-02	2.0864	0.2887E-03
100	0.186E+00	0.944E-01 (96,29)	0.25695E-02	2.0858	0.2837E-03
150	0.194E+00	0.977E-01 (96,29)	0.22968E-02	2.0853	0.2559E-03
200	0.199E+00	0.996E-01 (95,29)	0.20632E-02	2.0865	0.2272E-03
250	0.202E+00	0.101E+00 (97,28)	0.18582E-02	2.0878	0.2026E-03
300	0.204E+00	0.102E+00 (97,28)	0.16774E-02	2.0888	0.1826E-03
350	0.206E+00	0.102E+00 (97,28)	0.15188E-02	2.0896	0.1661E-03

Table 2: Analysis of solution solved using multigrid, 129 by 129, M=99 & L=16

It is not possible to use this mesh with the FDMG code, which requires the meshsize on level k to be given by $2^k - 1$. Instead meshes between 129 by 129 and 17 by 17 are used with FDMG. The results are shown in Table 2 and show broad agreement between the two methods.

TEST PROBLEM TWO

This test problem, which appears in Venner (ref. 5), is solved on a single 129 by 129 grid and a multigrid where the finest grid is 129 by 129 and the coarsest grid is 17 by 17. Due to symmetry, only the nodes in the positive y direction are used. For this lightly loaded problem, the values of Moes dimensionless parameters are $M = 20$ and $L = 10$. This in turn gives $\lambda = 0.2$. The maximum Hertzian pressure, p_h, at this load is 0.58 GPa if $\alpha = 1.7 \times 10^{-8}$. The equivalent Hamrock and Dowson's dimensionless parameters with U fixed at 1.0×10^{-11} are $W = 1.8915 \times 10^{-7}$ and $G = 4729$.

This problem was solved using 300 multigrid V-cycles with the results recorded every 10 iterations as shown in Table 3. The corresponding entries, from a single grid for 1500 iterations recorded every 100 iterations, are shown in Table 4.

Its	Hcent	Hmin @ (I,J)	RMSRES	SumP	ERR(4,3)
10	0.444E+00	0.387E+00 (84 , 1)	0.3368E-02	1.6055	0.100E-01
20	0.246E+00	0.158E+00 (97 ,29)	0.3038E-02	2.1670	0.455E-02
30	0.349E+00	0.225E+00 (99 ,27)	0.2849E-02	2.1304	0.160E-02
40	0.380E+00	0.236E+00 (99 ,26)	0.2700E-02	2.1080	0.854E-03
50	0.400E+00	0.243E+00 (99 ,26)	0.2569E-02	2.1081	0.651E-03
60	0.417E+00	0.251E+00 (100,25)	0.2450E-02	2.1090	0.558E-03
70	0.429E+00	0.257E+00 (100,25)	0.2341E-02	2.1075	0.468E-03
80	0.439E+00	0.261E+00 (100,25)	0.2239E-02	2.1056	0.411E-03
90	0.447E+00	0.265E+00 (100,25)	0.2143E-02	2.1039	0.361E-03
100	0.454E+00	0.268E+00 (101,24)	0.2053E-02	2.1024	0.310E-03
120	0.464E+00	0.272E+00 (100,24)	0.1888E-02	2.1000	0.237E-03
140	0.472E+00	0.275E+00 (100,24)	0.1740E-02	2.0981	0.182E-03
160	0.478E+00	0.278E+00 (100,24)	0.1607E-02	2.0966	0.139E-03
180	0.483E+00	0.280E+00 (100,24)	0.1489E-02	2.0956	0.113E-03
200	0.487E+00	0.281E+00 (100,24)	0.1384E-02	2.0949	0.935E-04
250	0.494E+00	0.284E+00 (100,24)	0.1173E-02	2.0938	0.629E-04
300	0.499E+00	0.286E+00 (100,24)	0.1028E-02	2.0933	0.451E-04

Table 3: Test Problem Two solved using multigrid, 129 by 129, M=20 & L=10

Its	Hcent	Hmin @ (I,J)	RMSRES	SumP	ERROR
10	0.1058E+01	0.9588E+00 (107, 1)	0.3103E-01	2.2451	0.138E-01
100	0.5642E+00	0.4797E+00 (75 , 1)	0.2430E-02	1.4367	0.583E-02
200	0.2143E+00	0.1225E+00 (100,27)	0.2215E-02	2.0141	0.289E-02
300	0.3136E+00	0.1999E+00 (98 ,28)	0.2101E-02	2.1684	0.567E-03
400	0.3585E+00	0.2260E+00 (100,26)	0.2017E-02	2.1243	0.391E-03
500	0.3780E+00	0.2329E+00 (99 ,26)	0.1945E-02	2.1089	0.247E-03
600	0.3932E+00	0.2395E+00 (99 ,26)	0.1879E-02	2.1067	0.194E-03
700	0.4062E+00	0.2455E+00 (100,25)	0.1818E-02	2.1047	0.163E-03
800	0.4167E+00	0.2503E+00 (100,25)	0.1761E-02	2.1028	0.137E-03
900	0.4253E+00	0.2543E+00 (100,25)	0.1707E-02	2.1014	0.117E-03
1000	0.4326E+00	0.2577E+00 (100,25)	0.1656E-02	2.1004	0.102E-03
1500	0.4576E+00	0.2692E+00 (100,24)	0.1431E-02	2.0976	0.585E-04

Table 4: Test Problem Two solved on a single 129 by 129 grid, M=20 & L=10

Results. The values obtained after 1500 iterations on a single grid, shown in Table 4, for the central, labelled Hcent, and minimum, labelled Hmin, film thickness is achieved using 120 multigrid iterations as shown in Table 3. Thus 1500 single grid iterations correspond to about 120 multigrid iterations. For this problem, Venner (ref. 6) achieved 0.502 and 0.349 for Hcent and Hmin, respectively, using a grid of $\{(x,y) : -4.5 \leq x \leq 1.5, -3.0 \leq y \leq 3.0\}$. If convergence is based on the sum of pressures on the entire grid, labelled SumP, then the value obtained using a multigrid method is slightly better than that obtained using a single grid method. Although the change in solution from the finest grid, 129 by 129, and the grid just above it, labelled ERR(4,3) in Table 3, and the change in solution from one iteration to next on a single grid, labelled ERROR in Table 4, are evaluated differently, they both seem to suggest that the solution has converged to the order of 10^{-4}. Venner's results quote a value of ERR(4,3) of 0.122. The relative computation times on a SGI R4400 workstation for the two methods on this problem are 8:00:00 on a single grid for 1500 iterations and 7:15:00 for 300 multigrid V-cycles. The multigrid method thus provides a means of obtaining solutions with greater efficiency. One potential area of difficulty with the multigrid method is that if the coarsest multigrid cannot adequately represent the solution, then the method may exhibit convergence difficulties.

Contour line plots of the film thickness and pressure showing the formation of side-lopes and the spike region are shown in Figures 3 and 4, respectively. The cavitated region is clearly shown on the right side of Figure 4 and is preceded by the pressure spike region which can be seen more clearly in Figure 5.

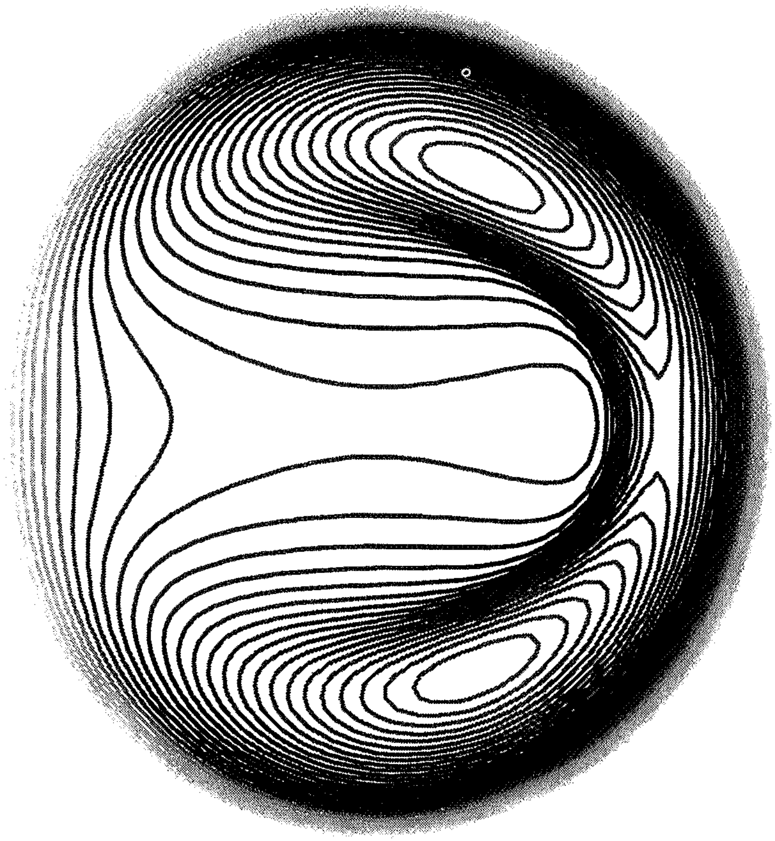


Figure 3: Contour line plot of film thickness on MG, 129 by 129, M=20 & L=10.

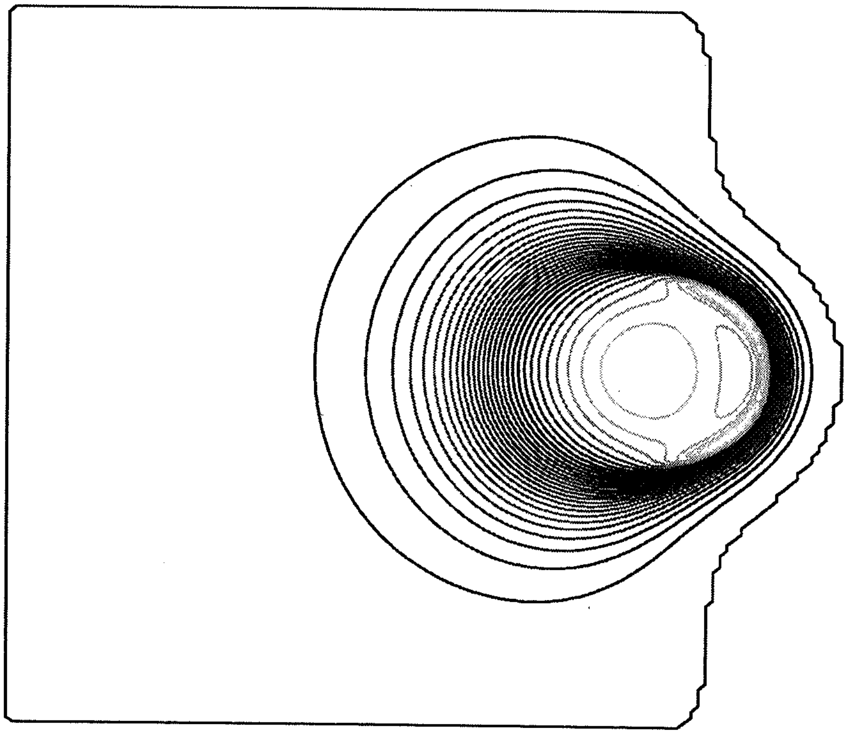


Figure 4: Contour line plot of pressure profile on MG, 129 by 129, M=20 & L=10.

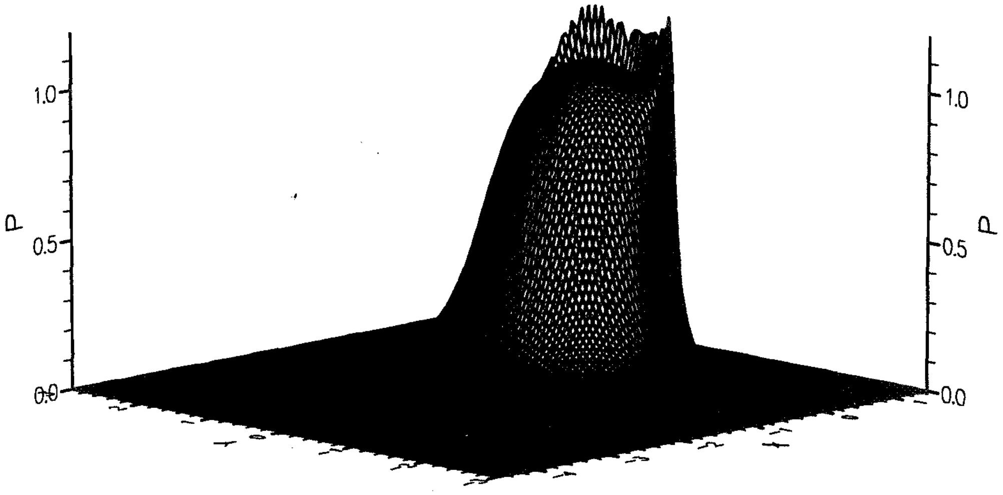


Figure 5: 3D pressure profile on MG, 129 by 129, M=20 & L=10.

CONCLUSIONS

The numerical results shown in this paper demonstrate how even a relatively standard multigrid code may be used to speed up the solution of EHL problems. The combination of the Effective Influence Method and multigrid method, which are both effective on their own, also appears to work well.

An outstanding issue concerns the treatment of convergence in EHL problems. From a practical engineering point of view it is the pressures and film thicknesses in the contact zone that are of interest and thus it is changes in these pressures which must tend to zero. The much larger residuals in the inlet region where the pressure is close to zero, though of potential cause of concern, may not influence the values of pressure in the contact region unduly. Furthermore, the Reynolds equation derivation is based on assumptions that are less valid in the inlet region. This is, however, an issue that needs to be further explored.

One possible way of obtaining a better understanding of the relationship between the residual and the solution is to compute error indicators in conjunction with adaptive meshes probably using a hierarchy of regular mesh patches to resolve the steep gradients in the pressure. It is this approach that will be our future research in this area.

ACKNOWLEDGMENTS

The authors would like to thank EPSRC and Shell Research Ltd, Thornton Research Centre for funding this work through an EPSRC CASE Studentships for E.N. Our colleagues at Leeds in the Mechanical Engineering Department D. Dowson, C. Taylor and D. Wang; J. Ware in Computer Studies and L. Scales at Shell are thanked for providing much constructive advice and support. Gareth Shaw of NAG Ltd is thanked for supplying the FDMG code.

REFERENCES

1. Gohar, R : Elastohydrodynamics. Ellis Horwood Limited, Chichester, England, 1988.

2. Hamrock, B. J.; and Dowson, D.: Isothermal Elastohydrodynamic Lubrication of Point Contacts, Part III, Fully Flooded Results. ASME Journal of Lubrication Technology, vol. 99, 1977, pp. 264-276.

3. Dowson, D.; and Wang, D.: An Analysis of the Normal Bouncing of a Solid Elastic Ball on an Oily Plate. Wear, vol. 179, 1994, pp. 29-37.

4. Venner, C. H.: Higher-Order Multilevel Solvers for the EHL Line and Point Contact Problem. Journal of Tribology, vol. 116, October, 1994, pp. 741-750.

5. Venner, C. H.; and Lubrecht, A. A.: Numerical Simulation of a Transverse Ridge in a Circular EHL Contact Under Rolling/Sliding. Journal of Tribology, vol. 116, October, 1994, pp. 751-761.

6. Venner, C. H.: Multilevel Solution of the EHL Line and Point Contact Problems. PhD. Thesis, University of Twente, The Netherlands. ISBN 90-9003974-0, 1991.

7. Dowson, D.; and Hamrock, B. J.: Numerical Evaluation of the Surface Deformation of Elastic Solids Subjected to a Hertzian Contact Stress. ASLE Trans., vol. 19, no. 4, 1976, pp. 279-286.

8. Wang, D.: Elastohydrodynamic Lubrication of Point Contacts for Layers of 'Soft' Solids and for 'Monolithic' 'Hard' Materials in the Transient Bouncing Ball Problems. PhD. Thesis, Department of Mechanical Engineering, University of Leeds, August, 1994.

9. Lubrecht, A. A.; Napel, W. E.; and Bosma, R.: Multigrid - An Alternative Method of Calculating Film Thickness and Pressure Profiles in Elastohydrodynamically Lubricated Line Contacts. ASME Journal of Tribology, vol. 108, no. 4, 1986, pp. 551-556.

10. Shaw, G. J.: FDMG Multigrid Software, version 3.0.

11. Hamrock, B. J.; and Dowson, D.: Isothermal Elastohydrodynamic Lubrication of Point Contacts, Part I, Theoretical Formulation. ASME Journal of Lubrication Technology, vol. 98, 1976, pp. 223-229.

MULTIGRID AND KRYLOV SUBSPACE METHODS FOR TRANSPORT EQUATIONS: ABSORPTION CASE

S. Oliveira
Computer Science Dept.
Texas A&M University
College Station, TX

September 22, 1995

SUMMARY

In this paper we look at Krylov subspace methods for solving the transport equations in a slab geometry. The spatial discretization scheme used is a finite element method called Modified Linear Discontinuous scheme (MLD). We investigate the convergence rates for a number of Krylov subspace methods for this problem and compare with the results of a spatial multigrid scheme.

INTRODUCTION

Transport equations describe the scattering and re-scattering of particles such as neutrons in a nuclear reactor, or light and infra-red radiation in the atmosphere. These equations are important, not only in nuclear engineering, but also in the study of the effects of greenhouse gases on the climate. A particularly important, although simple, model is of a single slab; this leads to integro-differential equations in one spatial variable and one angular variable. Unlike elliptic partial differential equations, these equations are based on highly non-normal operators, and require special care in their numerical treatment, especially for the regimes of physical interest: strong scattering, and weak or no absorption.

In the past decades there has been a great deal of work on numerical methods for large scale problems, such as partial differential equations. In this paper we focus on two of them: multigrid methods and Krylov subspace methods, as well as their application to transport equations.

In the past decade there has been an enormous development of Krylov subspace methods for non-symmetric and indefinite systems. These methods only require three operations to be available for their implementation: linear combinations, inner products, and matrix–vector products. Of these, it is assumed that matrix–vector products are the most complex to compute. As a result they can be efficiently implemented on scalar, vector and parallel computers.

These Krylov subspace methods that have been developed are all based on either the symmetric Lanczos, unsymmetric Lanczos, or Arnoldi methods for computing bases of Krylov subspaces. These include the CGS (Conjugate Gradient Squared) method, which is from the family of methods that uses the unsymmetric Lanczos method; the GMRES (Generalized Minimal RESidual) method, which uses the Arnoldi method; and LSQR (Least Squares/QR) approach, which uses the symmetric Lanczos method.

In addition, Krylov methods allow the easy incorporation of preconditioners. For solving $Ax = b$, a preconditioner is a matrix B, where Bu can be easily computed given a vector u and the system $BAx = Bb$ is easier to solve than the original system. Usually this is understood as finding B such that BA is a well conditioned matrix. Suitable matrices B can obtained by a number of different means. If A is "diagonally dominant", then B can be simply the inverse of the diagonal of A; other preconditioners are based on Gauss–Seidel or SOR iterations; another source is that of incomplete factorizations of sparse matrices, for example, ICCG, which combines incomplete Cholesky factorization with conjugate gradients. For a preconditioner to be incorporated into a Krylov subspace method, it is sufficient to use a routine to compute BAu for a given vector u by first computing $v = Au$ and then using a routine to compute Bv.

Another class of algorithms that has been extensively developed in the past decade are multigrid, or multilevel, algorithms. These have found a great deal of success in dealing with elliptic partial differential equations. Some multigrid methods have been developed for solving special cases of transport equations [5, 9, 10]. For one-dimensional problems, these can give exceptionally small convergence factors, and thus are extremely good methods [4, 5, 9, 10]. The development of parallel software for these methods is very time consuming due to the relaxation schemes used. For more general problems, and for two and three dimensional problems, the more "generic" Krylov subspace methods may be more suitable.

In this paper the usage of multigrid methods developed in [4, 5, 9] is investigated for the case of isotropic scattering with small but significant absorption. This case can lead to difficulties with the multigrid method given in [4, 5, 9], as is noted in

[6]. In [6] a modified algorithm is developed to handle the case with isotropic scattering; however here the Krylov subspace technique GMRES is used with the "pure scattering" multigrid algorithm to improve its performance and robustness.

TRANSPORT EQUATIONS

The description of the neutron transport problem is given in previous papers [1, 3, 5]. For steady state problems within the same energy group for the isotropic case (by isotropic we mean that the probability of scattering for the particles is the same for all directions), the transport equation in a slab geometry of slab width b becomes

$$\mu\frac{\partial\psi}{\partial x} + \sigma_t\psi = \frac{1}{2}\sigma_s\int_{-1}^{1}\psi(x,\mu')d\mu' + q(x,u), \tag{1}$$

for $x \in (0,b)$ and $\mu \in [-1,1]$. Here, $\psi(x,\mu)$ represents the flux of particles at position x traveling at an angle $\theta = \arccos(\mu)$ from the x-axis; $\sigma_t\,dx$, the expected number of interactions (absorptive or scattering) that a particle will have in traveling a distance dx; $\sigma_s\,dx$, the expected number of scattering interactions; $\sigma_a = \sigma_t - \sigma_s$, the expected number of absorptive interactions; and $q(x,\mu)$, the particle source. The boundary conditions prescribing particles entering the slab are

$$\psi(0,\mu) = g_0(\mu), \quad \psi(b,-\mu) = g_1(\mu), \tag{2}$$

for $\mu \in (0,1)$.

This problem is difficult for conventional methods to solve in two cases of physical interest:

1. $\gamma = \sigma_s/\sigma_t = 1$ (pure scattering, no absorption).
2. $1/\sigma_t \ll b$ (optically dense).

In fact, as $\sigma_t \to \infty$ and $\gamma \to 1$, the problem becomes singularly perturbed.

In this paper, the spatial discretization is a special finite element method called the Modified Linear Discontinuous (MLD) scheme (described in the next section), which behaves well in the thick limit.

In a previous paper this discretization has been solved by a multigrid algorithm [4]. This multigrid method was based on a two-cell red-black μ-line relaxation [5] with convergence factors of order $O((1/\sigma_t h)^2)$ when $\sigma_t h \gg 1$, and $O((\sigma_t h)^3)$ when $\sigma_t h \ll 1$.

Note that these multigrid operators are non-symmetric. Thus if they are used to precondition a Krylov subspace method, it must be a non-symmetric method such as GMRES, CGS, or QMR. In this paper we focus on GMRES.

DISCRETIZATION

The angular discretization is accomplished by expanding the angular dependence in Legendre polynomials, and is known as the S_N approximation when the first N Legendre polynomials are used. This results in a semidiscrete set of equations that resemble collocation at N Gauss quadrature points, μ_j, $j = 1, \ldots, N$, with weights w_j, $j = 1, \ldots, N$. Since the quadrature points and weights are symmetric about zero, we reformulate the problem in terms of the positive values, μ_j, $j = 1, \ldots, n$, where $n = N/2$. We define $\psi_j^+ = \psi(x, \mu_j)$ and $\psi_j^- = \psi(x, -\mu_j)$ for $j = 1, \ldots, n$. The spatial discretization is accomplished by the MLD scheme, which uses elements that are linear across each cell and discontinuous in the upwind direction. In our grid representation, the variable $\psi_{ij}^{+(-)}$ denotes the flux of particles at position x_i in the direction μ_j $(-\mu_j)$. The nodal equations are

$$\frac{\mu_j}{\sigma_t} \frac{\psi^+_{i+\frac{1}{2},j} - \psi^+_{i-\frac{1}{2},j}}{h_i} + \psi^+_{i,j} = \gamma \sum_{k=1}^{n} \omega_k (\psi^+_{i,k} + \psi^-_{i,k}) + q^+_{i,j}, \tag{3}$$

$$2\frac{\mu_j}{\sigma_t} \frac{\psi^+_{i+\frac{1}{2},j} - \psi^+_{i,j}}{h_i} + \psi^+_{i+\frac{1}{2},j} = \gamma \sum_{k=1}^{n} \omega_k (\psi^+_{i+\frac{1}{2},k} + 2\psi^-_{i,k} - \psi^-_{i-\frac{1}{2},k}) + q^+_{i,j}, \tag{4}$$

$$\frac{\mu_j}{\sigma_t} \frac{\psi^-_{i-\frac{1}{2},j} - \psi^-_{i+\frac{1}{2},j}}{h_i} + \psi^-_{i,j} = \gamma \sum_{k=1}^{n} \omega_k (\psi^+_{i,k} + \psi^-_{i,k}) + q^-_{i,j}, \tag{5}$$

and

$$2\frac{\mu_j}{\sigma_t} \frac{\psi^-_{i-\frac{1}{2},j} - \psi^-_{i,j}}{h_i} + \psi^-_{i-\frac{1}{2},j} = \gamma \sum_{k=1}^{n} \omega_k (\psi^-_{i-\frac{1}{2},k} + 2\psi^+_{i,k} - \psi^+_{i+\frac{1}{2},k}) + q^-_{i,j}, \tag{6}$$

$j = 1, \ldots, n$, $i = 1, \ldots, m$, with boundary conditions

$$\psi^+_{\frac{1}{2},j} = g^+_{0,j}, \quad \psi^-_{m+\frac{1}{2},j} = g^-_{1,j}, \tag{7}$$

$j = 1, \ldots, n$.

In our model, $x_{i+\frac{1}{2}}$ and $x_{i-\frac{1}{2}}$ are cell edges, $x_i = \frac{1}{2}(x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}})$ is the cell center, and $h_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ is the cell width, $1 \leq i \leq m$. Equations (3) and (5) are called balance equations and (4) and (6) are called edge equations. In block matrix form equations (3) – (7) can be written respectively as

$$B_i(\underline{\psi}^+_{i+\frac{1}{2}} - \underline{\psi}^+_{i-\frac{1}{2}}) + \underline{\psi}^+_i = \gamma R(\underline{\psi}^+_i + \underline{\psi}^-_i) + \underline{q}^+_i, \tag{8}$$

$$2B_i(\underline{\psi}^+_{i+\frac{1}{2}} - \underline{\psi}^+_i) + \underline{\psi}^+_{i+\frac{1}{2}} = \gamma R(\underline{\psi}^+_{i+\frac{1}{2}} + 2\underline{\psi}^-_i - \underline{\psi}^-_{i-\frac{1}{2}}) + \underline{q}^+_{i+\frac{1}{2}}, \tag{9}$$

$$B_i(\underline{\psi}^-_{i-\frac{1}{2}} - \underline{\psi}^-_{i+\frac{1}{2}}) + \underline{\psi}^-_i = \gamma R(\underline{\psi}^+_i + \underline{\psi}^-_i) + \underline{q}^-_i, \tag{10}$$

$$2B_i(\underline{\psi}^-_{i-\frac{1}{2}} - \underline{\psi}^-_i) + \underline{\psi}^-_{i-\frac{1}{2}} = \gamma R(\underline{\psi}^-_{i-\frac{1}{2}} + 2\underline{\psi}^+_i - \underline{\psi}^+_{i+\frac{1}{2}}) + \underline{q}^-_{i-\frac{1}{2}}, \tag{11}$$

$$\underline{\psi}^+_{\frac{1}{2}} = \underline{g}^+_0, \quad \underline{\psi}^-_{m+\frac{1}{2}} = \underline{g}^-_1, \tag{12}$$

$i = 1, \ldots, m$. Here,

$$B_i = \begin{bmatrix} \mu_1/\sigma_t h_i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_n/\sigma_t h_i \end{bmatrix}, \quad \text{and} \quad R = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \omega_1 & \cdots & \omega_n \end{bmatrix}, \tag{13}$$

where $\mu_1, \mu_2, \ldots, \mu_n$ are the positive Gauss quadrature points, $w_1, w_2, \ldots, w_n$ are the Gauss quadrature weights, and $\underline{\psi}^{+(-)}_i$ is an n-vector: $\underline{\psi}^{+(-)}_i = (\psi^{+(-)}_{i1}, \ldots, \psi^{+(-)}_{in})^T$.

In the computational grid, the inflow for positive angles is on the left, and the inflow for the negative angles is on the right of the whole domain. Figure 1 shows the computational domain with $2m+1$ spatial points and n angles. For a cell μ-line relaxation the inflows of each cell are assumed known. For a μ-line relaxation for the whole domain only the boundary conditions are assumed known.

Consider cell i. In one-cell μ-line relaxation cell, centers $\underline{\psi}^+_i$ and $\underline{\psi}^-_i$, together with the outflow variables $\underline{\psi}^-_{i-\frac{1}{2}}$ and $\underline{\psi}^+_{i+\frac{1}{2}}$, will be updated using the following matrix equation:

$$A\underline{u}_i = \underline{rhs}^1_i + \underline{rhs}^2_i \tag{14}$$

where the matrix A is given by

$$\begin{bmatrix} I + 2B_i - \gamma R & -2\gamma R & -2B_i & \gamma R \\ 0 & I - \gamma R & -\gamma R & B_i \\ B_i & -\gamma R & I - \gamma R & 0 \\ \gamma R & -2B_i & -2\gamma R & I + 2B_i - \gamma R \end{bmatrix} \tag{15}$$

with

$$\underline{u}_i = (\underline{u}^-_{i-\frac{1}{2}}, \underline{u}^+_i, \underline{u}^-_i, \underline{u}^+_{i+\frac{1}{2}}),$$

$$\underline{rhs}^1_i = (0, B_i\underline{u}^+_{i-\frac{1}{2}}, B_i\underline{u}^-_{i+\frac{1}{2}}, 0),$$

and

$$\underline{rhs}^2_i = (\underline{q}^-_{i-\frac{1}{2}}, \underline{q}^+_i, \underline{q}^-_i, \underline{q}^+_{i+\frac{1}{2}}).$$

Solving this matrix equation corresponds to performing a μ-line relaxation for one cell. To solve this system for all variables we consider the cells coupled together; thus, we have a μ-line relaxation for the whole domain, which we solve for all variables at each iterative step.

The linear system for the whole domain can be written as

$$\begin{bmatrix} D & C & & & & \\ B & D & C & & & \\ & B & D & C & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & C \\ & & & & B & D \end{bmatrix} \begin{bmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \vdots \\ \underline{u}_m \end{bmatrix} = \begin{bmatrix} \underline{rhs}_1^1 \\ \underline{rhs}_2^1 \\ \vdots \\ \vdots \\ \underline{rhs}_m^1 \end{bmatrix} + \begin{bmatrix} \underline{rhs}_1^2 \\ \underline{rhs}_2^2 \\ \vdots \\ \vdots \\ \underline{rhs}_m^2 \end{bmatrix} \tag{16}$$

where

$$\underline{rhs}^1 = (0, B_1\underline{\psi}^+_{1-\frac{1}{2}}, 0, 0, 0, 0, B_m\underline{\psi}^-_{m+\frac{1}{2}}, 0)^T. \tag{17}$$

KRYLOV SUBSPACE METHODS

After the discretizations are chosen for equations (1–2) the problem becomes one of finding the best methods for the solution of $Ax = b$, where A is a $q \times q$ matrix and x and b are vectors of size q. In these methods, iterative solutions of the form $x^{k+1} = x^k + p^k$ are constructed, where $p^k \in K_k(A, r^0)$. $K_k(A, r^0)$ is the Krylov space of dimension k, where $k \leq q$ and is defined as the span of $r^0, Ar^0, A^2r^0, \ldots, A^{k-1}r^0$.

The basic conjugate gradient algorithm of Hestenes and Stiefel [2] for symmetric positive matrices minimizes the residual in the A^{-1} norm ($\|x\|_{A^{-1}} = \sqrt{x^TA^{-1}x}$) over $K_k(A, r^0)$. After q steps, without roundoff errors, it zeros the residual. For nonsymmetric matrices this method does not work.

In this paper the solution of non-symmetric discretizations is investigated. Thus we must consider other Krylov methods. In addition we investigate the use of multigrid methods as preconditioners.

Krylov subspace methods are based on either the symmetric or unsymmetric Lanczos methods, or the Arnoldi method, applied either to A or to a closely related matrix. The symmetric Lanczos and Arnoldi algorithms generate (in exact arithmetic) orthonormal bases for $K_k(A, r^0)$, while the unsymmetric Lanczos produces a pair of biorthonormal bases for $K_k(A, r^0)$ and $K_k(A^T, r^0)$, respectively. In both cases the Lanczos methods produce a tridiagonal matrix that represents the original matrix on the Krylov subspaces, while the Arnoldi method produces a Héssenberg matrix that represents the matrix on the Krylov subspace $K_k(A, r^0)$. The unsymmetric Lanczos process is fast, but can suffer from numerical instability, known as breakdown. There are variants of these based on the look-ahead Lanczos algorithm, which is a stabilized version of the unsymmetric Lanczos method.

One of the most commonly used non-symmetric Krylov subspace solvers is GMRES. This method minimizes the residual over all solution vectors of the form $x^0 + p^k$ where p^k lies in $K_k(A, r^0)$.

MULTIGRID

To illustrate the multigrid scheme we consider it in the form of two grid levels. We use the notation h to indicate a fine grid and $2h$ to indicate a coarse grid, although our grids are not really assumed to be uniform. Let L^h denote the fine grid operator; L^{2h}, the coarse grid operator; and I_{2h}^h and I_h^{2h}, the interpolation and restriction operators, respectively. Let ν_1 and ν_2 be small integers (e.g., $\nu_1 = \nu_2 = 1$), which determine the number of relaxation sweeps performed before and after the coarse grid correction. Then one multigrid $V(\nu_1, \nu_2)$ cycle is represented (in two-grid form) by the following:

1. Relax ν_1 times on $L^h u^h = f^h$.
2. Calculate the residual $r^h = f^h - L^h u^h$.
3. Solve approximately $L^{2h} u^{2h} = I_h^{2h} r^h$.
4. Replace $u^h \leftarrow u^h + I_{2h}^h u^{2h}$.
5. Relax ν_2 times on $L^h u^h = f^h$.

The coarse grid operator, L^{2h}, is defined as

$$L^{2h} = I_{2h}^h L^h I_h^{2h}.$$

For the isotropic scattering the multigrid scheme was applied with regard to the spatial variable in [4, 5].

Figure 1 illustrates grid points on the fine grid and on the coarse grid. The interpolation and restriction operators for our previous multigrid schemes for transport equations were defined in [4, 5]. The L^h operator is given in (15). The coarse grid operator L^{2h} has the same form as L^h, but on the new grid.

NUMERICAL RESULTS

The numerical results presented here are for the isotropic transport equations, both with and without absorption. The methods used are mostly the multigrid method of [4, 5, 9] for isotropic transport equations without absorption, by itself, and this method used as a preconditioner for GMRES. The methods were implemented using the Meschach matrix library in C [12] and were run on a Sun SPARC 20. The

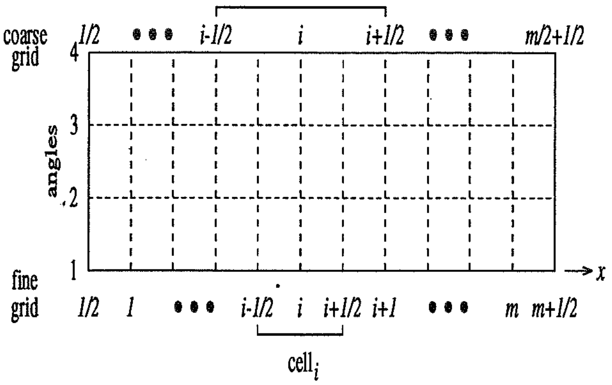


Figure 1: Computational Grid

test problems used had 64, 256, or 1024 cells, 16 angles; $\sigma_t h$ has the values 10^1, 10^2, 10^3, and 10^4, under several different regimes for $\gamma = \sigma_s/\sigma_t$. These absorption regimes are $\gamma = 1 - 1/(\sigma_t h)^2$, $\gamma = 1 - 1/(\sigma_t h)^3$, $\gamma = 0.99$, and $\gamma = 1$ (no absorption). The size of the test problems range from 4096 unknowns to 65 536 unknowns; σ_t ranges from 640 to 1.024×10^7.

The convergence factors were estimated for randomly generated solutions. The convergence factor estimate was obtained by taking the geometric average of the ratios of the norms of the residuals obtained from the last 5 iterations for each method, except where roundoff error caused the residual norm to plateau.

Note that in the tables an entry of the form $0.xxx(\pm y)$ means $0.xxx \times 10^{\pm y}$.

The convergence factor estimates are given in Table 1 ($\gamma = 1 - 1/(\sigma_t h)^2$), Table 2 ($\gamma = 1 - 1/(\sigma_t h)^3$), Table 3 ($\gamma = 0.99$), and Table 4 ($\gamma = 1$). The first regime is both of physical interest and also is the more difficult to solve using the standard MLD discretization and the simple interpolation and restriction operators. This corresponds

# cells	method	$\sigma_t h$			
		10^1	10^2	10^3	10^4
64	MG	0.262	0.736	0.885	0.931
	MG+GMRES	0.0487	0.191	0.236	0.129
256	MG	0.263	0.741	0.900	0.952
	MG+GMRES	0.0477	0.208	0.550	0.213
1024	MG	0.263	0.741	0.905	0.950
	MG+GMRES	0.0454	0.208	0.695	0.219

Table 1: Convergence factors for $\gamma = 1 - 1/(\sigma_t h)^2$

		$\sigma_t h$			
# cells	method	10^1	10^2	10^3	10^4
64	MG	0.481	0.647	0.266	0.046
	MG+GMRES	0.101	0.0319	0.677(−2)	0.176(−2)
256	MG	0.486	0.844	0.722	0.344
	MG+GMRES	0.124	0.165	0.0559	0.0110
1024	MG	0.488	0.896	0.895	0.780
	MG+GMRES	0.122	0.484	0.255	0.0848

Table 2: Convergence factors for $\gamma = 1 - 1/(\sigma_t h)^3$

		$\sigma_t h$			
# cells	method	10^1	10^2	10^3	10^4
64	MG	0.262	0.0530	0.111(−2)	0.121(−4)
	MG+GMRES	0.0487	0.215(−2)	0.186(−4)	0.221(−6)
256	MG	0.263	0.0530	0.111(−2)	0.121(−4)
	MG+GMRES	0.0477	0.285(−2)	0.190(−4)	0.216(−6)
1024	MG	0.263	0.0530	0.111(−2)	0.121(−4)
	MG+GMRES	0.0454	0.279(−2)	0.189(−4)	0.222(−6)

Table 3: Convergence factors for $\gamma = 0.99$

		$\sigma_t h$			
# cells	method	10^1	10^2	10^3	10^4
64	MG	0.320(−4)	0.206(−6)	0.119(−5)	0.116(−3)
	MG+GMRES	0.681(−5)	0.710(−7)	0.196(−6)	0.987(−5)
256	MG	0.323(−4)	0.207(−6)	0.187(−4)	0.189(−2)
	MG+GMRES	0.105(−4)	0.910(−7)	0.354(−5)	0.135(−3)
1024	MG	0.324(−4)	0.299(−5)	0.303(−3)	0.0321
	MG+GMRES	0.160(−4)	0.649(−6)	0.230(−4)	0.182(−2)

Table 4: Convergence factors for $\gamma = 1$

to a situation in which the scattered particles undergo a large number of scatterings; in addition they have a significant probability of being absorbed in a cell, and also of "escaping" a cell. The numerical difficulty of the problem is clearly evident in the convergence factors obtained.

Results for diverse Krylov subspace methods using diagonal and ILU (Incomplete LU factorization) preconditioning are reported in [11], but they were only obtained for relatively small values of $\sigma_t h$. These methods do not seem adequate for the very large values of $\sigma_t h$ that are studied here. For example, there it is reported that the convergence factor for 100 cells, 4 angles, and $\sigma_t h = 1$, using GMRES with an ILU preconditioner, was 0.705 and clearly deteriorates as the number of cells and $\sigma_t h$ increase. In contrast, with the multigrid method either used directly or as a preconditioner, the convergence factor for 256 cells, 16 angles, and $\sigma_t h = 10$ was 0.734×10^{-3} in the "no absorption" case.

The worst regime for absorption is that with $\gamma = 1 - 1/(\sigma_t h)^2$. In this regime, deterioration in the rates of convergence for both the direct multigrid and the GMRES/multigrid methods is evident. Nevertheless, with GMRES, the convergence rates are significantly faster and would give overall rates of convergence at least twice as fast and up to nearly a factor of 30 faster. Since each step of GMRES only requires one matrix-vector multiplication for the operator and for the preconditioner and has negligible overhead, preconditioning would give improved overall speed. The multigrid methods of [6] appear to give much better convergence factors, but at the cost of additional complexity of the algorithm, not to mention the additional effort needed to perform the analysis to design the correct operators for handling this case.

Outside this regime, the GMRES/multigrid algorithm works consistently better than the direct multigrid algorithm, and where the original multigrid algorithm performs well, the GMRES/multigrid algorithm improves the convergence factor by a factor of as much as 100. However, in these cases it would only roughly halve the number of iterations needed to achieve a small error tolerance. As noted for the most difficult regime, where the original multigrid algorithm has difficulty, using it as a preconditioner for GMRES gives much better results.

CONCLUSIONS

In this paper the multigrid method for isotropic transport equations of [4, 5, 9] for the "no absorption" case is applied to problems with absorption both as a pure iterative method and as a preconditioner for GMRES. In all cases, GMRES improves the convergence factor, although the value of this appears to be much greater for the cases in which the nonabsorption multigrid algorithm has difficulty (such as the absorption regime $\gamma = 1 - 1/(\sigma_t h)^2$). The multigrid algorithm thus has been demonstrated as an efficient preconditioner for GMRES. Together they are robust and, in addition, work well for the absorption regime. We expect multigrid methods to work well for the other Krylov subspace methods which we have used, such as CGS, LSQR, and CGNE, for which preconditioning is essential.

ACKNOWLEDGMENTS

I would like to thank Dora Lazarova for the first implementation of the codes, and Dr. David Stewart for helping with the use of Meschach.

REFERENCES

[1] V. Faber and T. A. Manteuffel. *Neutron transport from the viewpoint of linear algebra*, Lecture Notes in Pure and Applied Mathematics, April 1989.

[2] M. R. Hestenes and E. Stiefel. *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Standards Vol. 49, pp. 409–436, 1952.

[3] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*, John Wiley and Sons, New York, 1984.

[4] T. Manteuffel, S. McCormick, J. Morel, S. Oliveira, and G. Yang. *Fast Multigrid Solver for Transport Problems I: Pure Scattering*, to appear in SIAM J. of Sci. Computing, Vol. 16, No. 3, May 1995.

[5] T. Manteuffel, S. McCormick, J. Morel, S. Oliveira and G. Yang. *A Parallel Version of a Multigrid Algorithm for Isotropic Transport Equations*, SIAM J. of Sci. Computing Vol. 15, pp. 474–493, 1994.

[6] T. Manteuffel, S. McCormick, J. Morel and G. Yang. *Fast Multigrid Solver for Transport Problems with Absorption*, Technical Report, University of Colorado at Denver, 1993.

[7] J. E. Morel and E. W. Larson. *A New Class of S_N Spatial Differencing Schemes*, Nucl. Sci. Eng., 1988.

[8] S. Oliveira. *Parallel Multigrid Methodos for Transport Equations: Anisotropic Case*, submitted to Parallel Computing.

[9] S. Oliveira. *Parallel Multilevel Methods For Transport Equations*, Ph.D. thesis, Mathematics Department, University of Colorado at Denver, May 1993.

[10] S. Oliveira. *Parallel Multilevel Algorithms for Anisotropic Transport Equations, Proceedings CTAC93 conference*, Publ. World Scientific, Singapore, pp. 388–396, 1994.

[11] S. Oliveira. *Krylov Subspace Methods for Transport Equations*, Proceedings XENFIR III, Aguas de Lindóia, Brazil, 1995.

[12] D. E. Stewart and Z. Leyk. *Meschach: Matrix Computations in C*, Proceedings of the Centre for Mathematical Sciences #32, Australian National University, Canberra, 1994.

FAST MULTIGRID TECHNIQUES IN TOTAL VARIATION–BASED IMAGE RECONSTRUCTION

Mary Ellen Oman
Department of Mathematical Sciences
Montana State University
Bozeman, MT [1]

SUMMARY

Existing multigrid techniques are used to effect an efficient method for reconstructing an image from noisy, blurred data. Total Variation minimization yields a nonlinear integro-differential equation which, when discretized using cell-centered finite differences, yields a full matrix equation. A fixed point iteration is applied with the intermediate matrix equations solved via a preconditioned conjugate gradient method which utilizes multi-level quadrature (due to Brandt and Lubrecht) to apply the integral operator and a multigrid scheme (due to Ewing and Shen) to invert the differential operator. With effective preconditioning, the method presented seems to require $\mathcal{O}(n)$ operations. Numerical results are given for a two-dimensional example.

INTRODUCTION

The problem of reconstructing an image from noisy, blurred data can be represented by the model equation

$$z = Ku + \epsilon, \tag{1}$$

where K is a smoothing operator, ϵ is noise, and u is to be recovered from noisy data z. K is typically a Fredholm first kind integral operator, $(Ku)(x) = \int k(x,y)u(y)dy$, which is compact, so problems of this form are ill-posed; i.e., small perturbations in the data will produce wildly varying u's.

In the past, attempts to apply multigrid techniques to inverse problems similar to this have produced rather disappointing results. Either multigrid has been applied directly to (1) without stabilization (see [1] as an example) which produces poor quality reconstructions for high noise-to-signal ratios (due to the ill-posedness of the problem), or stabilization has been applied, but multigrid displays slow convergence (see [2]). In this paper it will be demonstrated how to overcome these difficulties with existing multigrid tools, obtaining a fast algorithm to approximate u in (1).

To stabilize problem (1) Tikhonov regularization, or penalized least squares, is used:

$$\min T_\alpha(u), \quad \text{where } T_\alpha(u) = \frac{1}{2}\|Ku - z\|^2{}_2 + \alpha J(u), \tag{2}$$

[1] Research was supported in part by a DOE-EPSCoR graduate fellowship.

where α is a positive parameter, and J is a known functional.

A common choice for J is

$$J(u) = \int_\Omega |\nabla u|^2, \tag{3}$$

but this assumes $u \in H^1(\Omega)$. Hence, it is unsuitable for image processing applications, where one wants to recover sharp edges, i.e., discontinuous u.

In their seminal paper on Total Variation-based denoising [3], Osher, Rudin, and Fatemi considered the functional

$$J(u) = \int_\Omega |\nabla u|. \tag{4}$$

To overcome difficulties associated with nondifferentiability at $\nabla u = 0$, consider the modification

$$J_\beta(u) = \int_\Omega \sqrt{|\nabla u|^2 + \beta dx}, \quad \beta \geq 0. \tag{5}$$

For $\beta = 0$, J_β is the total variation of u. Figure 1 (excerpted from [4]) depicts a comparison of reconstructions of u in (2). In subplot B, J is as in (3), hence the reconstruction is smooth; in subplot C, $J = J_\beta$ as in (5), and a blocky image is recovered; and subplot D shows a filtered Fourier reconstruction of the data. Clearly Total Variation produces a superior reconstruction in this test case.

Minimizing T_α as given in (2) with J defined as in (5) yields the nonlinear integro-differential equation

$$K^*(Ku - z) + \alpha \nabla J_\beta(u) = 0 \text{ for } x \in \Omega, \quad \text{and } \frac{\partial u}{\partial n} = 0 \text{ for } x \in \partial\Omega. \tag{6}$$

This can be written in operator form as

$$\tilde{K}u + \alpha L(u)u = K^*z, \tag{7}$$

where

$$\tilde{K} = K^*K \tag{8}$$

and $L(u)$ is the diffusion operator whose action on a function v is given by

$$L(u)v = -\nabla \cdot \left(\frac{1}{\sqrt{|\nabla u|^2 + \beta}} \nabla v \right). \tag{9}$$

Note that both $\tilde{K}$ and $L(u)$ are symmetric positive semidefinite operators.

The following fixed point algorithm [4] can then be applied to handle the nonlinearity:

$$(\tilde{K} + \alpha L(u^{(\nu)}))u^{(\nu+1)} = K^*z, \quad \nu = 0, 1, \ldots \tag{10}$$

At each iteration it is necessary to solve a non-sparse linear system. This paper presents multigrid techniques for solving these systems efficiently.

The Denoising section deals with the case when K is the identity operator, the denoising problem. The Deconvolution section returns to the original problem (1) where

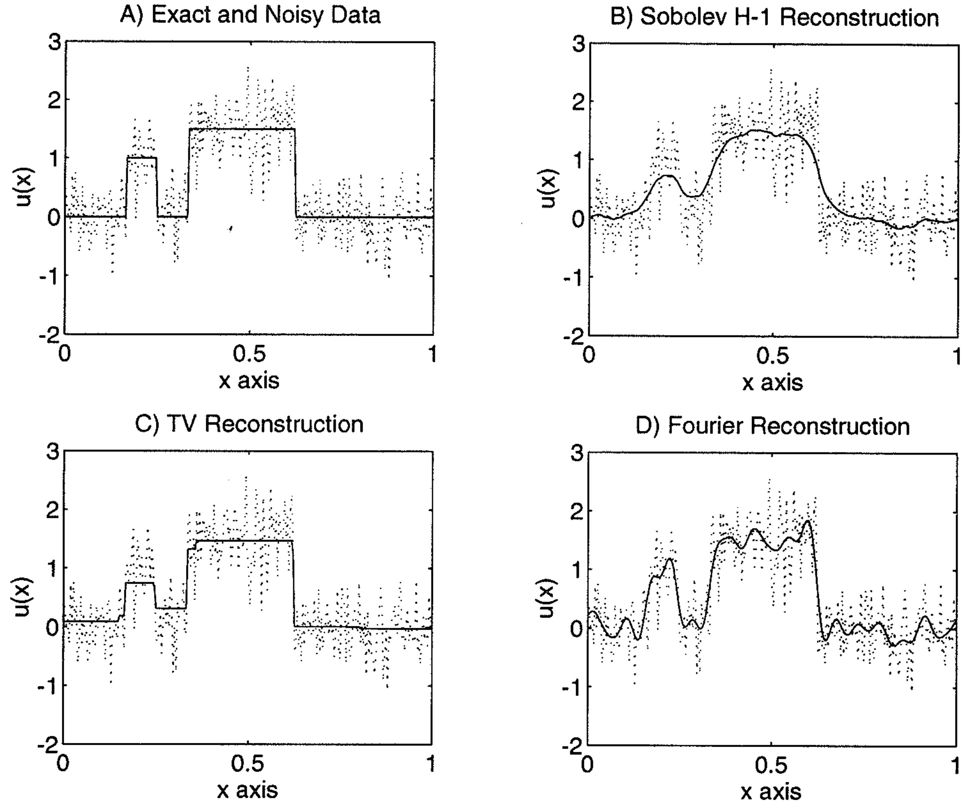


Figure 1: Denoised reconstructions obtained using various filtering techniques. Dotted lines represent noisy data. Solid line in subplot A is exact solution. Solid lines in subplots B-D are reconstructions.

K is a Fredholm first kind integral operator. Included are discussions of multi-level integration, preconditioning, and a recapitulation of the algorithm. The Numerical Results section discusses observed convergence rates for the numerical implementation and includes a two-dimensional example.

DENOISING

First, consider the case $Ku = u$. This corresponds to denoising an image, and (10) is reduced to

$$(1 + \alpha L(u^{(\nu)}))u^{(\nu+1)} = z, \quad \nu = 0, 1, \ldots . \tag{11}$$

At each iteration it is necessary to solve a linear diffusion equation whose diffusivity depends on the previous iterate $u^{(\nu)}$. This iteration is referred to as a "lagged diffusivity fixed point iteration," and is denoted here as FP (see [4] for details).

Note that the diffusion coefficient $1/\sqrt{|\nabla u|^2 + \beta}$ is poorly behaved where ∇u is large. The cell-centered finite difference discretization [5] is applied to overcome this

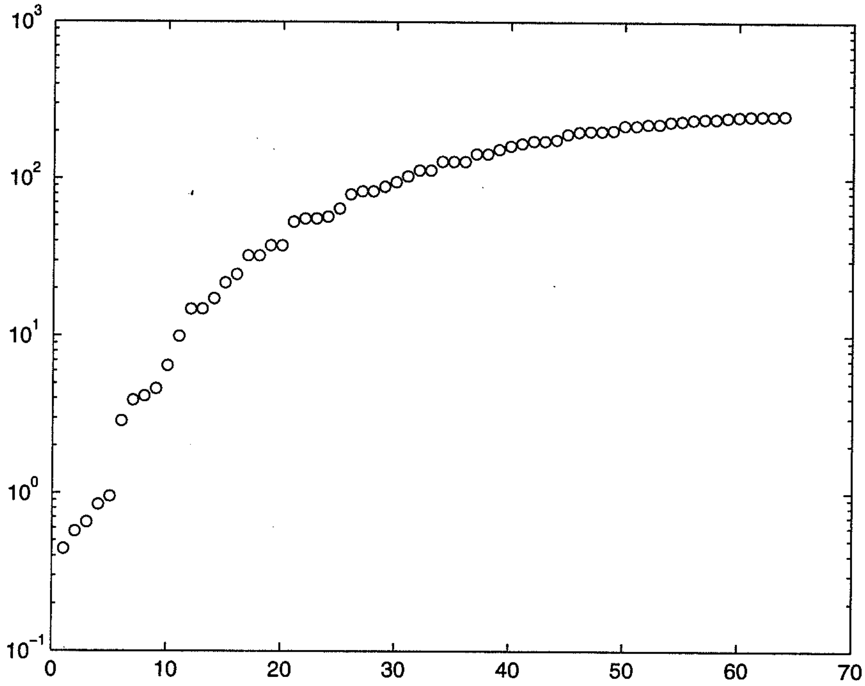


Figure 2: The spectrum of the discretized operator $I + \alpha L(u)$ for a fixed u in one space dimension.

difficulty. After discretization, one must solve a sparse, block tridiagonal matrix equation to obtain $u^{(\nu+1)}$ at each fixed point iteration. Figure 2 shows the spectrum of the operator from (11) for a fixed u. A preconditioned conjugate gradient method has been employed with a multigrid preconditioner developed by Ewing and Shen [5].

DECONVOLUTION

Now consider the case when K is a Fredholm first kind integral operator. The matrix obtained from the discretization of $\tilde{K} + \alpha L(u^{(\nu)})$ in (10) is no longer sparse. Hence, to use the lagged diffusivity fixed point iteration as before, a full matrix equation must be solved for each iteration. The conjugate gradient method can again be applied but with a cost of n^2 operations per iteration. In typical 2-D image processing applications $n^2 \approx 10^{12}$; clearly this operation count is unacceptable. The Multi-level integration section describes a scheme for reducing the complexity of one conjugate gradient iteration from n^2 to n.

In [6], Brandt and Lubrecht describe a method based on multigrid techniques for approximately evaluating $\tilde{K}v$ which requires only $\mathcal{O}(n)$ operations. The general idea is

$$\tilde{K}v \approx \tilde{K}^h v^h \approx \underbrace{\Pi^h_H}_{\mathcal{O}(n)} \underbrace{\tilde{K}^H}_{\mathcal{O}(N^2)} \underbrace{(\Pi^H_h v^h)}_{\mathcal{O}(n)} . \tag{12}$$

Here h and n indicate the mesh spacing and number of nodes on the fine grid, and similarly, H and N indicate the coarse grid with $N << n$. Π^h_H and Π^H_h are coarse-to-fine and fine-to-coarse intergrid transfer operators, respectively.

To evaluate $\tilde{K}^h v^h$ cheaply, restrict v^h to the coarse grid, apply the coarse grid operator $\tilde{K}^H$ at a cost of $\mathcal{O}(N^2)$ operations, and then interpolate $\tilde{K}^H v^H$ back to the fine grid.

To see the details of this approximation, choose q^{th} order transfer operators: Π^h_H, a coarse-to-fine mesh transfer (interpolation), and Π^H_h, a fine-to-coarse mesh transfer (restriction). Using p^{th} order quadrature, the operation becomes

$$\begin{array}{rcl} [\tilde{K}v](x^H_I) & = & \int_0^1 \tilde{k}(x^H_I, y)v(y)dy, \qquad I = 1, \ldots, N \\ \\ & = & h\sum_{j=1}^n \tilde{k}(x^H_I, x^h_j))v^h_j + \mathcal{O}(h^p) \\ \\ & = & h\sum_{j=1}^n [\tilde{k}(x^H_I, x^H_{\cdot})\Pi^h_H]_j v^h_j + \mathcal{O}(h^p) + \mathcal{O}(H^q) \\ \\ & = & h\sum_{J=1}^N \tilde{k}(x^H_I, x^H_J)[(\Pi^h_H)^T v^h]_J + \mathcal{O}(h^p) + \mathcal{O}(H^q) \end{array} \tag{13}$$

Then $[\tilde{K}v](x^H_I)$ can be interpolated to the fine grid by Π^h_H with $\mathcal{O}(H^q)$ accuracy. The entire application looks like

$$\tilde{K}^h v^h = \Pi^h_H \tilde{K}^H (\Pi^h_H)^T v^h + \mathcal{O}(h^p) + \mathcal{O}(H^q). \tag{14}$$

If $N^2 \approx n$ then $H^q \approx h^p$, provided $q = 2p$, and this calculation requires only $\mathcal{O}(n)$ operations and maintains $\mathcal{O}(h^p)$ accuracy. To see this, let $n = 2^{lev}$ ($lev > 0$ is the number of levels, or nested grids), let $n+1$ be the number of points in the finest mesh with spacing $h = \frac{1}{n}$, and let the coarsest mesh have $N+1$ points with spacing $H = \frac{1}{N}$ where $N = 2^{lev/2}$. With second order quadrature ($p = 2$), $K^H \Pi^H_h v^h$ can be calculated in $\mathcal{O}(N^2) = \mathcal{O}((2^{lev/2})^2) = \mathcal{O}(n)$ operations. Fourth order transfer operators ($q = 4$) ensure that the accuracy of $\Pi^h_H \tilde{K}^H \Pi^H_h v^h$ is $\mathcal{O}(h^2) + \mathcal{O}(H^4) = \mathcal{O}(h^2)$. Note that $\Pi^H_h = c(\Pi^h_H)^T$, with $c = H/h$; hence, $\Pi^h_H \tilde{K}_H (\Pi^h_H)^T$ is symmetric.

This provides an $\mathcal{O}(n)$ method of applying $\tilde{K}$ which maintains $\mathrm{O}(h^2)$ accuracy. Hence, an iteration of the conjugate gradient method applied to the system (10) will use only $\mathcal{O}(n)$ operations. However, $\tilde{K} + \alpha L(u)$ is not typically well-conditioned. The top right subplot of figure 4 depicts the eigenvalues of this operator for a fixed

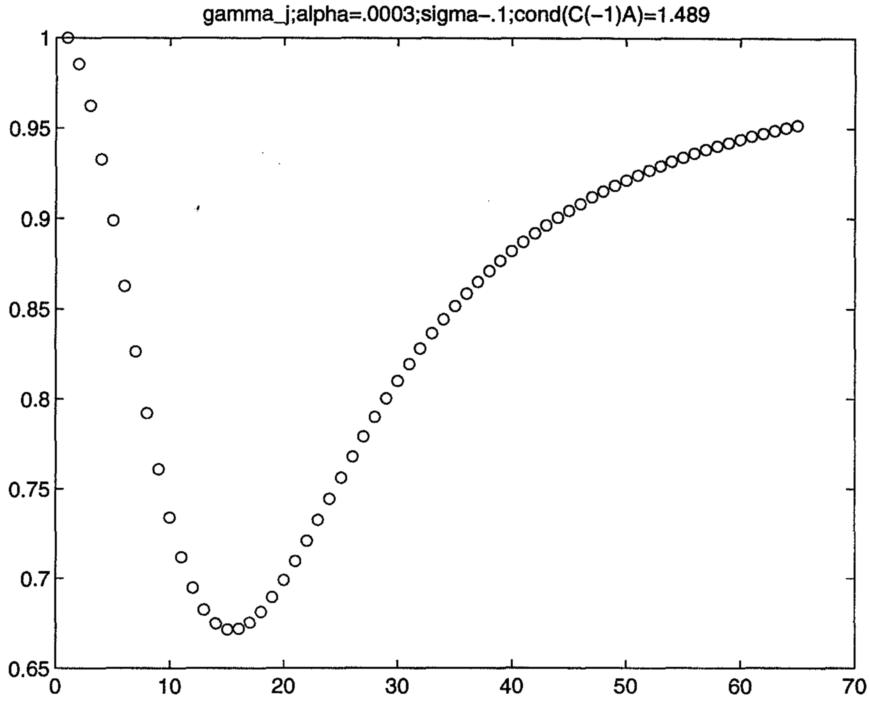


Figure 3: Eigenvalues of the preconditioned matrix, $C^{-1/2}AC^{-1/2}$ where $Lu = -\nabla^2 u$, $C = bI + \alpha L$ and b is the maximum eigenvalue of $\tilde{K}$.

u, α, and β. Note that these eigenvalues range over three orders of magnitude. Preconditioning must be used to improve convergence.

Preconditioning

To simplify notation, define

$$A \stackrel{\text{def}}{=} \tilde{K} + \alpha L(u) \tag{15}$$

For insight into the choice of a preconditioner, consider the 1-D case on $0 \le x \le 1$ with $L(u)$ replaced by the negative Laplacian and periodic boundary conditions, where K is a convolution operator, $Ku = \int_0^1 k(x-y)u(y)dy$, with Gaussian convolution kernel, $k(x) = \sqrt{\frac{2}{\pi}}e^{-x^2/\sigma^2}$. Then L has eigenvalues $j^2\pi^2$ which tend to ∞, $\tilde{K}$ has eigenvalues $\frac{2}{\pi}e^{-\sigma^2 j^2/2}$ which tend to 0, and L commutes with $\tilde{K}$.

This eigenvalue structure suggests a preconditioner of the form $C = bI + \alpha L$. Then the iteration matrix becomes $C^{-1/2}AC^{-1/2} = C^{-1}A$ with eigenvalues

$$\gamma_j = \frac{\frac{2}{\pi}e^{-\sigma^2 j^2/2} + \alpha\pi^2 j^2}{b + \alpha\pi^2 j^2}, j = 1, 2, \ldots \tag{16}$$

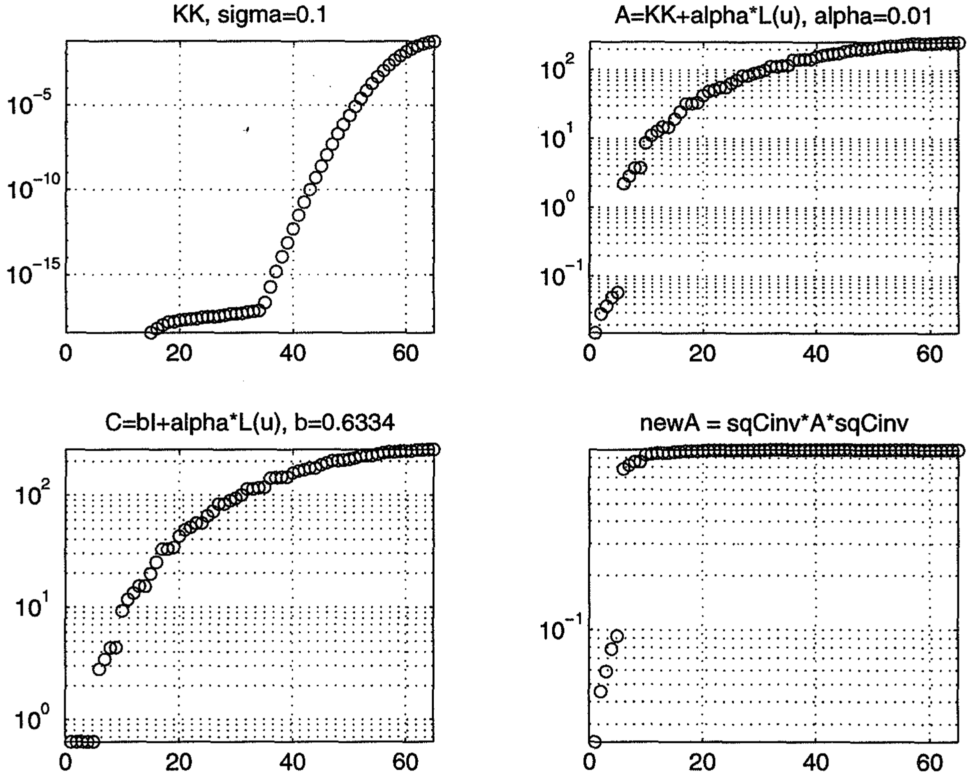


Figure 4: Eigenvalues of the discretized operator matrices $\tilde{K} = K^*K$, A, C, and $C^{-1/2}AC^{-1/2}$ where K is a convolution operator with kernel $k(x) = \left(\frac{2}{\pi}\right)^{1/2} e^{-x^2/\sigma^2}$, $C = bI + \alpha L(u)$, and $L(u)$ is the nonlinear operator as in FP.

The γ_j tend to 1 as $j \to \infty$ independent of b. To ensure $\gamma_j \approx 1$ for small values of j, choose the largest eigenvalue of $\tilde{K}$ for b,

$$b = \rho(\tilde{K}) = \frac{2}{\pi} e^{-\dot{\sigma}^2/2}. \tag{17}$$

Figure 3 shows the eigenvalues of the iteration matrix $C^{-1}A$, which result from this choice of b. Notice that $cond(C^{-1}A) \approx 1$. This implies that the conjugate gradient method will converge very rapidly.

With the more general diffusion operator defined in (9), this choice of b is yet reasonable as shown in Figure 4. Here, the eigenvalues of the matrices A, C, and $C^{-1/2}AC^{-1/2}$ are shown. Although the eigenvalues are not all near one as in the constant diffusivity case, there is still clustering at one. The "stray" eigenvalues correspond to jump discontinuities in u. Thus, $C = bI + \alpha L(u)$ is an effective preconditioner for this case as well.

A fast reconstruction algorithm

The fixed point iteration and preconditioned conjugate gradient techniques described above can be combined to form an efficient reconstruction algorithm. What follows is the outline of such an algorithm for the two-dimensional deconvolution problem. This algorithm is used to obtain the numerical results presented in the following section.

- Apply fixed point iteration as in (10).

- To solve $(\tilde{K}+\alpha L(u^{(\nu)}))u^{(\nu+1)} = K^*z$, apply a preconditioned conjugate gradient method with preconditioner $C = bI + \alpha L(u^{(\nu)})$ with $b = \rho(\tilde{K})$.

- Within the preconditioned conjugate gradient method, use multi-level integration for each application of $\tilde{K}$.

- Within each iteration of the preconditioned conjugate gradient method, solve equations of form $Cv = (bI + \alpha L)v = f$ by a preconditioned conjugate gradient method with the Ewing-Shen multigrid preconditioner [5].

Notice that $C = bI + \alpha L$ is essentially the same as the operator in a fixed point iteration of the denoising problem (11). The multi-level integration is $\mathcal{O}(n)$ as shown above and in [6]. Therefore, the complexity of the preconditioned conjugate gradient method to solve $(\tilde{K} + \alpha L(u^{(\nu)}))u^{(\nu+1)} = K^*z$ depends on the complexity of solving $Cv = f$.

NUMERICAL RESULTS

In Figures 5 and 6, the operator K has been taken to be a convolution integral operator with kernel,

$$k(x) = \left(\frac{2}{\pi}\right)^{1/2} e^{-x^2/\sigma^2} \tag{18}$$

as in the Multi-level Integration section. Figure 5 presents convergence results for this 2-D example with noise-to-signal ratio $= 1$ and kernel-width parameter, $\sigma = 0.075$. Subplot A depicts the norms of the differences between successive iterates. Subplot B shows the norm of the gradient of T_α as in (6). Subplot C plots the preconditioned conjugate gradient convergence factor for each fixed point iteration where the geometric mean convergence factor is calculated by

$$\text{convergence factor} = \exp\left[\frac{1}{M}\sum_{m=1}^{M} \ln(\frac{res^{m+1}}{res^{m}})\right] \tag{19}$$

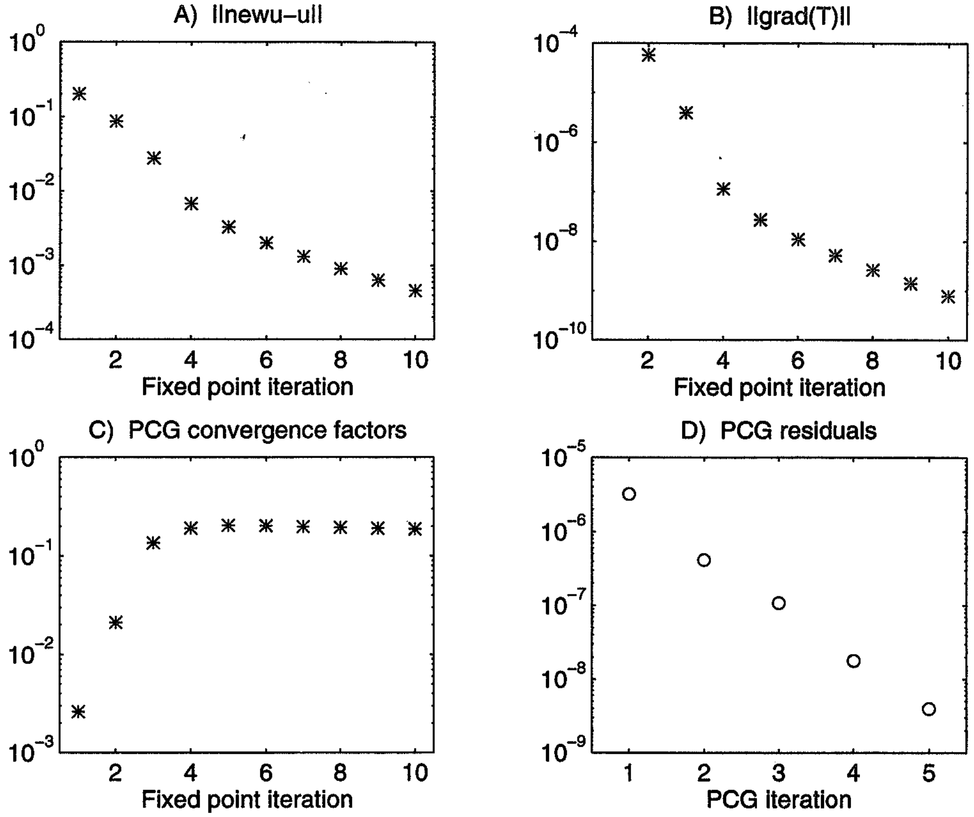


Figure 5: Subplots A and B show the norms of the differences between iterates and the gradient of the function T_α, respectively. Subplot C contains the convergence history of the preconditioned conjugate gradient method with preconditioner $C = bI + \alpha L$ at each fixed point iteration. Subplot D plots the residuals of the preconditioned conjugate gradient method for 5 iterations at the tenth fixed point iteration.

where res^m is the residual calculated at the $(m-1)^{\text{st}}$ preconditioned conjugate gradient iterate. Subplot D records the norms of the residuals at each preconditioned conjugate gradient iteration for the tenth fixed point iteration. Figure 6 shows the noisy data (with noise-to-signal ratio = 1), $z = Ku_{\text{exact}} + \epsilon$ and the subsequent reconstruction obtained by the above algorithm.

These results show that the described algorithm can be used to obtain reconstructions even for very noisy data. The convergence of the preconditioned conjugate gradient method is quite fast as evidenced by Figure 5, Subplots C and D. It is known that the multi-level integration method has $\mathcal{O}(n)$ complexity. Hence, the complexity of the preconditioned conjugate gradient method to solve $(\tilde{K}+\alpha L(u^{(\nu)}))u^{(\nu+1)} = K^*z$ depends on the complexity of solving $Cv = f$, where $C = bI+\alpha L(u^{(\nu)}))$. This system is nearly identical to the one obtained in the discretization of the denoising problem, and for the results given here the same solver has been used, i.e., a preconditioned con-

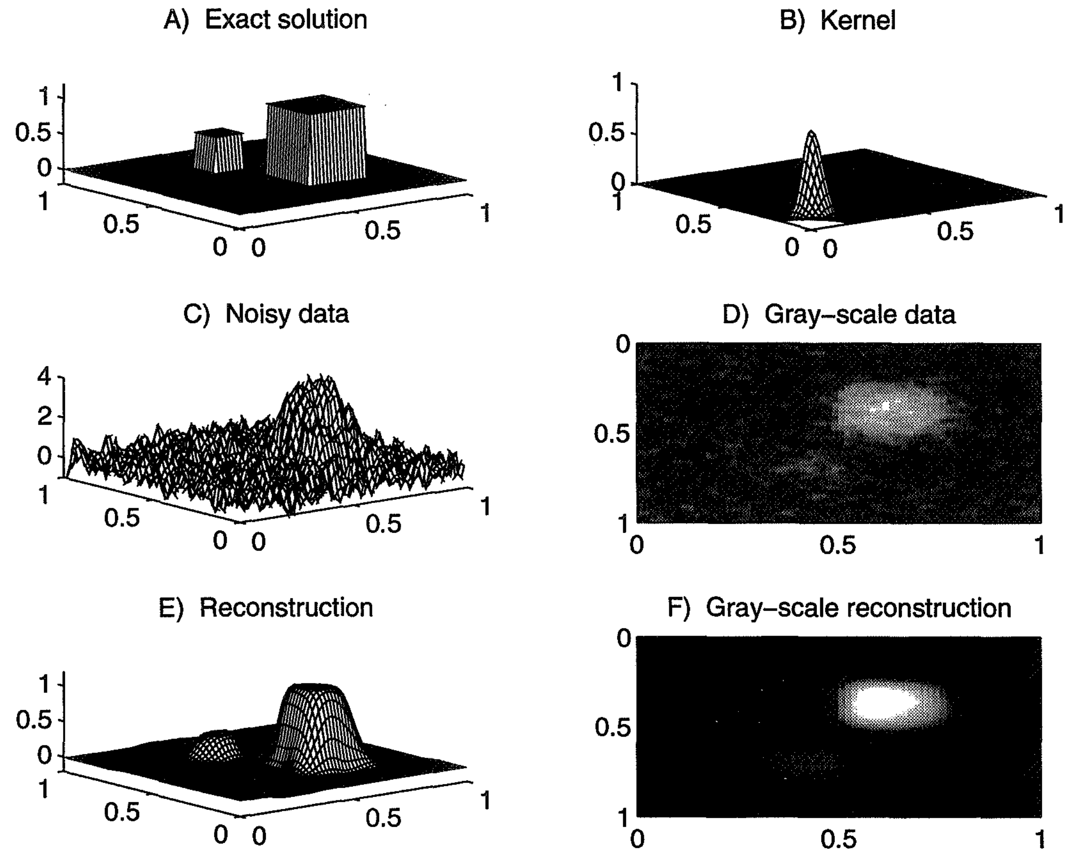


Figure 6: Subplot A shows the exact solution. Subplot B shows the kernel of the convolution operator. Subplots C and D show the data with added noise (noise-to-signal ratio = 1). Subplots E and F show the subsequent reconstruction with the algorithm described.

jugate gradient with a cell-centered finite difference multigrid preconditioning step. This method appears to be nearly $\mathcal{O}(n)$ in complexity.

REFERENCES

[1] Zhou, K.; and Rushforth, C. K.: *Image Restoration Using Multigrid Methods*, Applied Optics, vol' 30, No. 20 (1991), pp. 2906-2912.

[2] McCormick, S. F.: *Multilevel Projection Methods for Partial Differential Equations*, CBMS-NSF Regional Conference Series in Applied Mathematics, No. 62 (1992), Section 4.1, pp. 62-70.

[3] Rudin, L. I.; Osher, S.; and Fatemi, E.: *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D, vol 60 (1992), pp. 259-268.

[4] Vogel, C. R.; and Oman, M. E.: *Iterative Methods for Total Variation Denoising*, SIAM Journal of Scientific Computing, vol 17, No. 1 (1996).

[5] Ewing, R. E.; and Shen, J.: *A multigrid algorithm for the cell-centered finite difference scheme*, in the Proceeding of the 6th Copper Mountain Conference on Multigrid Methods, April 1993.

[6] Brandt, A.; and Lubrecht, A. A.: *Multilevel Matrix Multiplication and Fast Solution of Integral Equations*, Journal of Computational Physics, vol 90 (1990), pp. 348-370.

Page intentionally left blank

A MULTILEVEL ALGORITHM FOR THE SOLUTION OF SECOND ORDER ELLIPTIC DIFFERENTIAL EQUATIONS ON SPARSE GRIDS

Christoph Pflaum
Institut für Informatik, Technische Universität München
D-80290 München, Germany

SUMMARY

A multilevel algorithm is presented that solves general second order elliptic partial differential equations on adaptive sparse grids. The multilevel algorithm consists of several V-cycles. Suitable discretizations provide that the discrete equation system can be solved in an efficient way. Numerical experiments show a convergence rate of order $O(1)$ for the multilevel algorithm.

1 Introduction

In 1990, Bungartz and Zenger used hierarchical bilinear finite elements on a sparse grid to discretize Poisson's equation on the unit square (see [1] and [2]). The discrete equation system was solved by a recursive algorithm. Balder extended this idea for the solution of the Helmholtz equation in the d-dimensional space (see [3]).

In this paper, a multilevel algorithm is presented, that solves general second order elliptic partial differential equations on adaptive sparse grids. This multilevel algorithm consists of several V-cycles in one direction and of a Gauss-Seidel relaxation on each level. The restrictions of these V-cycles are a semicoarsening. Thus, the multilevel algorithm is similar to the multilevel algorithm in [4] and [5]. The Gauss-Seidel relaxation and the restriction and prolongation is made like the multilevel projection method in [6]. The multilevel cycle of the sparse grid multilevel algorithm is called Q-cycle. The problem of this Q-cycle is the calculation of the right hand side during the restriction. In case of general second order elliptic differential the exact stiffness matrix is so complicated that it is not possible to calculate the right hand side in an efficient way. This means that one multilevel cycle costs more than $O(N^2)$ operations, while $O(N \log N)$ is the number of sparse grid points. Thus, it is necessary to approximate the bilinear form a corresponding to the elliptic equation.

We studied two approximations of the bilinear form a. First, the variable coefficients in the bilinear form were replaced by a piecewise constant sparse grid interpolant. Then, it is possible to calculate the right hand side in an efficient way. But even an additional simplification of the bilinear form a is possible. For Laplace's equation some hierarchical basis functions are orthogonal with respect to the bilinear form corresponding to Laplace's equation. Therefore, it makes sense to replace the bilinear form a by a simplified bilinear form $\tilde{a}_h$, which has similar orthogonality properties even in case of general elliptic differential equations. This gives the discretization with semi-orthogonality (see section 3). A convergence with order $O(N^{-1} \log N)$ could be proved for this discretization of the Helmholtz equation (see [7]). Numerical experiments show the same behavior of convergence as for the original bilinear form even in case of more complicated elliptic differential equations. The advantage of the semi-orthogonality is that Q-cycle of the sparse grid multilevel algorithm becomes as simple as the V-cycle on full grids with bilinear finite elements. The reason for this is that nearly the same equations can be used for both multilevel cycles. On every level relaxations are made with a nine-point stencil. The restriction and the prolongation from one level to another one are made in the same way as in the case of full grids. For this it is only necessary to ignore the sparse grid points which are not contained in the actual level. This is allowed by the semi-orthogonality. All numerical experiments show a convergence rate with order $O(1)$ for the sparse grid multilevel algorithm. The multilevel algorithm requires only $O(N \log N)$ operations per cycle.

For simplicity, the discretizations and the algorithms in this paper are explained only for the regular sparse grids $\mathcal{D}_n$. However, it is possible to generalize the algorithms for adaptive sparse grids. The Q-cycle has been implemented for adaptive sparse grids and solves general second order elliptic differential equations.

Throughout the paper, it is $h = 2^{-n}$, where $n \in \mathbb{N}$ and $\Omega =]0,1[^2$.

2 Sparse Grids and Sparse Grid Interpolation

The set of one dimensional grid points is

$$\mathcal{P} = \left\{ \sum_{i=0}^{n} d_i \cdot \frac{1}{2^i} | n \in \mathbb{N}_0 \textit{ and } d_0 = 1,\ d_1 = -1,\ d_2, \ldots, d_n \in \{1, -1\} \right\} \cup \{0\}.$$

These points are illustrated in Figure 1. For every $x \in \mathcal{P} \backslash \{0\}$, there exist unique $n \in \mathbb{N}_0$ and $d_0 = 1$, $d_1 = -1$, $d_2, \ldots, d_n \in \{1, -1\}$ such that $x = \sum_{i=0}^{n} d_i \cdot \frac{1}{2^i}$. Therefore, we can define the depth of a point $x \in \mathcal{P}$ by

$$T(0) = 0 \qquad T(x) = n \quad \text{for } x \in \mathcal{P} \backslash \{0\}.$$

The regular sparse grids $\mathcal{D}_n$ and $\overset{\circ}{\mathcal{D}}_n$ are defined by

$$\begin{aligned}\mathcal{D}_n &:= \{(x,y) \in \mathcal{P} \times \mathcal{P} |\ T(x) + T(y) \leq n+1\}, \\ \overset{\circ}{\mathcal{D}}_n &:= \mathcal{D}_n \cap]0,1[^2,\end{aligned}$$

where $n \in \mathbb{N}$. A more detailed description of general abstract and adaptive sparse grids and their properties is given in [8] and [9].

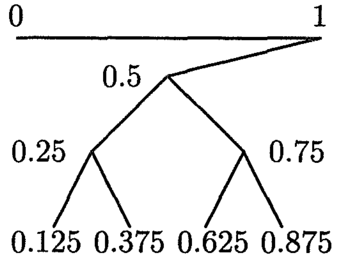
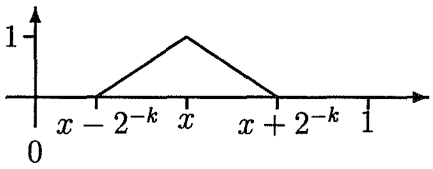


Figure 1. *Tree of possible grid points.*

Now, we will define the sparse grid interpolation with piecewise bilinear functions. For every $x \in \mathcal{P}$ and $k \in \mathbb{N}$, we define the piecewise linear function

$w_x^k : [0,1] \mapsto \mathbb{R}$ 

and for every $(x,y) \in \mathcal{P} \times \mathcal{P}$ and $k,l \in \mathbb{N}$ the piecewise bilinear function

$$\begin{aligned}v_{(x,y)}^{k,l} : \bar{\Omega} &\mapsto \mathbb{R} \\ v_{(x,y)}^{k,l}(x',y') &:= w_x^k(x') \cdot w_y^l(y').\end{aligned}$$

The hierarchical basis functions of the point $(x,y) \in \mathcal{P} \times \mathcal{P}$ is the function

$$v_{(x,y)} := v_{(x,y)}^{T(x),T(y)}. \tag{1}$$

There are two regular finite element spaces for the regular sparse grid $\mathcal{D}_n$

$$\begin{aligned}V_{\mathcal{D}_n} &:= \text{span}_{\mathbb{R}}\{v_z | z \in \mathcal{D}_n\} \subset W_2^1(\Omega) \cap \mathcal{C}(\bar{\Omega}) \qquad \text{and} \\ \overset{\circ}{V}_{\mathcal{D}_n} &:= \text{span}_{\mathbb{R}}\{v_z | z \in \overset{\circ}{\mathcal{D}}_n\} \subset \overset{\circ}{W}{}_2^1(\Omega) \cap \mathcal{C}(\bar{\Omega}).\end{aligned}$$

There is a unique sparse grid interpolation operator $\mathcal{I}_{\mathcal{D}_n} : \mathcal{C}(\bar{\Omega}) \mapsto V_{\mathcal{D}_n}$ such that $\mathcal{I}_{\mathcal{D}_n}(f)(z) = f(z) \quad \forall z \in \mathcal{D}_n$ (see [2]).

The sparse grid interpolation error with piecewise bilinear functions is now:

Theorem 1 (Sparse Grid Interpolation Error). *There exists a constant $C > 0$ such that the error in the W_2^1-norm is for $h = 2^{-n}$*

$$\begin{aligned}\|f - \mathcal{I}_{\mathcal{D}_n}(f)\|_{W_2^1} &\leq C\|f\|_{W_2^{G,4}} h && \textit{for } f \in W_2^{G,4}(\Omega), \quad \textit{and} \\ \|f - \mathcal{I}_{\mathcal{D}_n}(f)\|_{W_2^1} &\leq C\|f\|_{W_2^{G,3}} h \log h^{-1} && \textit{for } f \in W_2^{G,3}(\Omega),\end{aligned}$$

where

$$H^{G,l}(\Omega) := \{f \in L_2(\Omega)\, |\|f\|_{H^{G,l}} < \infty\} \ \textit{and}\ \|f\|_{H^{G,l}} := \left\| \left(\left\| \frac{\partial^{i+j} f}{\partial x^i \partial y^j} \right\|_{L_2} \right)_{i+j \leq l,\ i,j < \frac{l}{2}+1} \right\|_{l_2}$$

The proof of Theorem 1 is given in [1] and [7]. At the end of this section, we define the following full grids and a full grid finite element space

$$\Omega_{k,l} := \{(x,y) \in \mathcal{P} \times \mathcal{P} | T(x) \leq k \text{ and } T(y) \leq l\} \quad \text{and} \quad \overset{\circ}{\Omega}_{k,l} := \Omega_{k,l} \cap]0,1[^2$$
$$\text{and} \quad \overset{\circ}{V}^{k,l} := \text{span}_{\mathbb{R}}\{v_z | z \in \overset{\circ}{\Omega}_{k,l}\}.$$

3 Discretization of Elliptic Equations

We use the same notation as in [10]. Let $f \in (L_2(\Omega))'$ and

$$\begin{aligned} a : \overset{\circ}{W}{}^1_2(\Omega) \times \overset{\circ}{W}{}^1_2(\Omega) &\mapsto \mathbb{R} \\ (u,v) &\mapsto \int_\Omega \sum_{|\alpha|,|\beta|\leq 1} a_{\alpha,\beta}(D^\alpha u)(D^\beta v)\, d(x,y), \end{aligned}$$

where α, β are multiindices and $A = (a_{\alpha,\beta})_{|\alpha|,|\beta|\leq 1} \in \left(\mathcal{C}(\bar{\Omega})\right)^{3\times 3}$. Let us assume that a is continuous and $\overset{\circ}{W}{}^1_2(\Omega)$-elliptic. We are looking for a solution $u \in \overset{\circ}{W}{}^1_2(\Omega)$ of the equation

$$a(u,v) = f(v) \qquad \text{for all } \; v \in \overset{\circ}{W}{}^1_2(\Omega). \tag{1}$$

The problem is now that we cannot replace $\overset{\circ}{W}{}^1_2(\Omega)$ by the finite element space $\overset{\circ}{V}_{\mathcal{D}_n}$ and use the same bilinear form a. If we did so, we would get a manifold of stiffness matrices of dimension more than $O(2^n n)$ for this class of elliptic equations. Then, we would not be able to store the stiffness matrix in a sparse grid data structure. Therefore, we replace the bilinear form a by an approximate bilinear form. First, we replace a by

$$\begin{aligned} a_h : \overset{\circ}{W}{}^1_2(\Omega) \times \overset{\circ}{W}{}^1_2(\Omega) &\mapsto \mathbb{R} \\ (u,v) &\mapsto \int_\Omega \sum_{|\alpha|,|\beta|\leq 1} \mathcal{I}^c_{\mathcal{D}_n}(a_{\alpha,\beta})(D^\alpha u)(D^\beta v)\, d(x,y), \end{aligned}$$

where $\mathcal{I}^c_{\mathcal{D}_n}(a_{\alpha,\beta})$ is a suitable sparse grid interpolant. Second, we replace a_h by a bilinear form $\tilde{a}_h$ with a semi-orthogonality property. For the definition of the semi-orthogonality property, we need the set of pairs of semi-orthogonal grid points (see Figure 2)

$$\mathcal{O}_h := \mathcal{O}^l_h \cup \mathcal{O}^r_h,$$

where

$$\begin{aligned} \mathcal{O}^l_h &:= \{((x,y),(x',y')) \in \overset{\circ}{\mathcal{D}}_n \times \overset{\circ}{\mathcal{D}}_n | T(x) < T(x') \quad \textit{and} \quad T(y) > T(y') \quad \textit{and} \\ &\qquad \text{supp}(v_{(x,y)}) \cap \text{supp}(v_{(x',y')}) \cap \mathcal{D}_n = \emptyset \quad \textit{and} \\ &\qquad \text{supp}(v_{(x,y)}) \cap \text{supp}(v_{(x',y')}) \neq \emptyset\}, \\ \mathcal{O}^r_h &:= \{(z,z') \in \overset{\circ}{\mathcal{D}}_n \times \overset{\circ}{\mathcal{D}}_n | (z',z) \in \mathcal{O}^l_h\} \qquad \textit{and} \\ \text{supp}(v) &:= \{z \in \Omega | v(z) \neq 0\} \quad \textit{for} \quad v \in \mathcal{C}(\bar{\Omega}). \end{aligned}$$

Observe that here the support *supp* of a function is not compact in general.

Now we define the semi-orthogonality property of a bilinear form.

Definition 1 (Semi-Orthogonality Property).

A bilinear form $b : \overset{\circ}{V}_{\mathcal{D}_n} \times \overset{\circ}{V}_{\mathcal{D}_n} \mapsto \mathbb{R}$ *has the semi-orthogonality property, if*

$$b(v_z, v_{z'}) = 0 \quad \textit{for every} \quad (z, z') \in \mathcal{O}_h.$$

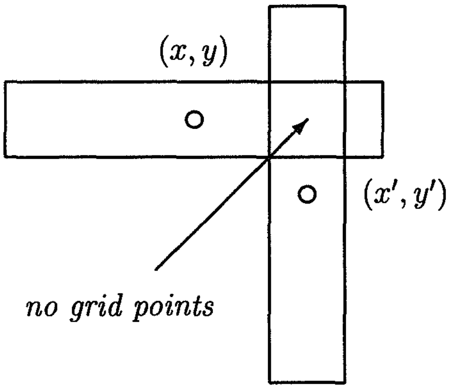


Figure 2. *Supports of Hierarchical Basis Functions of Semi-Orthogonal Grid Points.*

A simple calculation shows that the bilinear form $(w, v) \mapsto \int_\Omega \langle \nabla w, \nabla v \rangle d(x, y)$ has the semi-orthogonality property. In case of general second order elliptic differential equations, we define the discrete bilinear form $\tilde{a}_h$ by its values on the hierarchical basis

$$\begin{aligned} \tilde{a}_h : \overset{\circ}{V}_{\mathcal{D}_n} \times \overset{\circ}{V}_{\mathcal{D}_n} &\mapsto \mathbb{R} \\ \tilde{a}_h(v_z, v_{z'}) &:= \begin{cases} 0 & \text{for} \quad (z, z') \in \mathcal{O}_h \\ a_h(v_z, v_{z'}) & \text{for} \quad (z, z') \notin \mathcal{O}_h. \end{cases} \end{aligned}$$

Obviously, $\tilde{a}_h$ has the semi-orthogonality property. The discretization of equation (1) with semi-orthogonality is now:

DISCRETIZATION WITH SEMI-ORTHOGONALITY *Find a* $\tilde{u}_h \in \overset{\circ}{V}_{\mathcal{D}_n}$ *such that*

$$\tilde{a}_h(\tilde{u}_h, v_h) = f(v_h) \quad \textit{for all} \quad v_h \in \overset{\circ}{V}_{\mathcal{D}_n}.$$

4 Multilevel Algorithm

Let $\hat{a}_h$ be the bilinear form a_h or $\tilde{a}_h$. We want to solve the following problem:

Discrete Equation System *Find $u_h \in \overset{\circ}{V}_{\mathcal{D}_n}$ such that*

$$\hat{a}_h(u_h, v_h) = f(v_h) \qquad \forall v_h \in \overset{\circ}{V}_{\mathcal{D}_n}. \tag{1}$$

Assume that $\tilde{u} \in \overset{\circ}{V}_{\mathcal{D}_n}$ is an approximate solution of the discrete equation system. Obviously, there are $\lambda_z \in \mathbb{R}$ such that $\tilde{u} = \sum_{z \in \overset{\circ}{\mathcal{D}}_n} \lambda_z v_z$. For fixed $k, l \in \mathbb{N}$, $k + l \leq n + 1$, we make the following decomposition:

$$\tilde{u} = \tilde{u}^{k,l} + \tilde{u}^{k,l}_{rest},$$

where

$$\tilde{u}^{k,l} = \sum_{z \in \overset{\circ}{\Omega}_{k,l}} \lambda_z v_z \in \overset{\circ}{V}^{k,l} \quad \text{and} \quad \tilde{u}^{k,l}_{rest} = \sum_{\substack{z = (x,y) \in \overset{\circ}{\mathcal{D}}_n \,\wedge \\ (T(x) > k \,\vee\, T(y) > l)}} \lambda_z v_z$$

For the construction of a multilevel algorithm, we have to push $\tilde{u}^{k,l}_{rest}$ to the right hand side. Thus, we define

$$f^{k,l}(v_h) := f(v_h) - \hat{a}_h(\tilde{u}^{k,l}_{rest}, v_h) \quad \text{for } v_h \in \overset{\circ}{V}^{k,l}. \tag{2}$$

Now, we can define the

Equation System of Level (k, l) *Find $\tilde{u}^{k,l} \in \overset{\circ}{V}^{k,l}$ such that*

$$\hat{a}_h(\tilde{u}^{k,l}, v_h) = f^{k,l}(v_h) \qquad \forall v_h \in \overset{\circ}{V}^{k,l} \tag{3}$$

Naturally, if $\tilde{u} = u$ is the exact solution of the discrete equation system, then $\tilde{u}^{k,l}$ is the solution of the equation system of level (k, l). If $\tilde{u}^{k,l}$ is the solution of the equation system of level (k, l) for every $k + l \leq n + 1$, then $\tilde{u} = u$ is the exact solution of the discrete equation system.

For relaxations, it is helpful to form the equation system of level (k, l) in a matrix equation. Therefore, we define the vectors $\left(U^{k,l}_z\right)_{z \in \overset{\circ}{\Omega}_{k,l}}, \left(F^{k,l}_z\right)_{z \in \overset{\circ}{\Omega}_{k,l}} \in \mathbb{R}^{|\overset{\circ}{\Omega}_{k,l}|}$ and the matrix $\left(A^{k,l}_{z,z'}\right)_{z,z' \in \overset{\circ}{\Omega}_{k,l}} \in \mathbb{R}^{|\overset{\circ}{\Omega}_{k,l}| \times |\overset{\circ}{\Omega}_{k,l}|}$ by

$$\begin{aligned} \tilde{u}^{k,l} &= \sum_{z \in \overset{\circ}{\Omega}_{k,l}} U^{k,l}_z v^{k,l}_z && (4) \\ F^{k,l}_z &:= f^{k,l}(v^{k,l}_z) \quad \text{and} && (5) \\ A^{k,l}_{z,z'} &:= \hat{a}_h\left(v^{k,l}_{z'}, v^{k,l}_z\right). && (6) \end{aligned}$$

The following matrix equation is equivalent to the equation system of level (k, l):

Matrix Equation of Level (k, l)
Find $\left(U_z^{k,l}\right)_{z \in \mathring{\Omega}_{k,l}} \in \mathbb{R}^{|\mathring{\Omega}_{k,l}|}$ *such that*

$$A^{k,l} U^{k,l} = F^{k,l} \tag{7}$$

Now, we want to construct a multilevel algorithm. The principal data to be stored are:

- k, l: depth of the actual level. 2^{-k} and 2^{-l} are the mesh sizes of the full grid $\mathring{\Omega}_{k,l}$ corresponding to the actual level,
- $(U_z)_{z \in \mathring{\mathcal{D}}_n}$: the actual approximate solution,
- $(F_z)_{z \in \mathring{\mathcal{D}}_n}$: the right hand side of the actual level, and
- $(W_z)_{z \in \mathring{\mathcal{D}}_n}$: the one dimensional hierarchical surplus in the direction of the last restriction.

First, we have to define a relaxation step in the level (k, l). Let

$$\tilde{u}_{old} = \tilde{u}_{old}^{k,l} + \tilde{u}_{old,rest}^{k,l}$$

be the decomposition of the actual approximate solution. Assume $U_z = \tilde{u}_{old}^{k,l}(z)$ for all $z \in \mathring{\Omega}_{k,l}$.

Procedure: Relaxation
Choose $(U_z)_{z \in \mathring{\Omega}_{k,l}}$ *for the start solution of (7). Make a standard relaxation step (e.g. Gauss-Seidel-relaxation) of equation (7). This gives the new approximate solution* $(U_z)_{z \in \mathring{\Omega}_{k,l}}$ *on the level* (k, l)*.*

After one relaxation step, we define $\tilde{u}_{new}^{k,l}(z) := U_z$ for all $z \in \mathring{\Omega}_{k,l}$. $\tilde{u}_{new}^{k,l} \in \mathring{V}^{k,l}$ is the new approximate solution on the level (k, l). The new approximate solution is now

$$\tilde{u}_{new} := \tilde{u}_{new}^{k,l} + \tilde{u}_{old,rest}^{k,l}.$$

But after one relaxation, we only have $\tilde{u}_{new}(z) = U_z$ for all $z \in \mathring{\Omega}_{k,l}$. For the propagation of $\tilde{u}_{new}$ to other grid points, we need the procedures *restriction* and *prolongation*. The procedure *prolongation* calculates $\tilde{u}_{new}$ on the new level.

Procedure: Restriction in x-direction
For $(x, y) \in \mathring{\Omega}_{k,l}$ *with* $T(x) = k$ *do*

$$W_{(x,y)} := U_{(x,y)} - 0.5 * \left(U_{(x+2^{-k},y)} + U_{(x-2^{-k},y)}\right);$$

Procedure: Prolongation in x-direction
For $(x,y) \in \mathring{\Omega}_{k,l}$ with $T(x) = k$ do

$$U_{(x,y)} := W_{(x,y)} + 0.5 * (U_{(x+2^{-k},y)} + U_{(x-2^{-k},y)});$$

The procedures *Restriction in y-direction* and *Prolongation in y-direction* are defined analogous.

The procedures *Restriction* and *Prolongation* calculate

$$U_z := \tilde{u}_{new}(z) \quad \text{for} \quad z \in \mathring{\Omega}_{k_{new},l_{new}},$$

where (k_{new}, l_{new}) is the new level. The procedures *Restriction* and *Prolongation* can do this only if the multilevel algorithm satisfies the following rule:

Restriction-Prolongation-Rule
Assume that *Restriction in x-direction* was used from the level (k', l') to the level $(k'-1, l')$. Then use *Prolongation in x-direction* with $k = k' - 1$ next time only if $l = l'$.
Assume that *Restriction in y-direction* was used from the level (k', l') to the level $(k', l'-1)$. Then use *Prolongation in y-direction* with $l = l' - 1$ next time only if $k = k'$.

Last, we need the procedure

Procedure: Calculation of the right hand side
This procedure calculates $F_z := F_z^{k,l}$ for all grid points $z \in \mathring{\Omega}_{k,l}$.

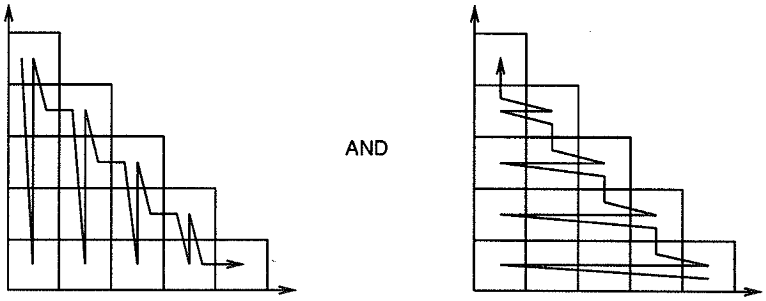


Figure 3: Q-Cycle of the multilevel algorithm on a sparse grid

Now we can explain the *Q-cycle* (see Figure 3):

```
THE Q-CYCLE {
    Step 1: Way in x-direction
    LET $k := 1$;
    WHILE $k < n$ {
        Step 1.1: V-cycle in one direction
        LET $l := n - k + 1$;
        WHILE $l > 1$ {
            Restriction in y-direction; AND $l := l - 1$;
            Calculate the right hand side;
        }
        Relaxation;
        WHILE $l < n - k + 1$ {
            $l := l + 1$ AND Prolongation in y-direction;
            Calculate the right hand side;
            Relaxation;
        }
        Step 1.2: Changing $k$
        Restriction in y-direction; AND $l := n - k$;
        Calculate the right hand side;
        $k := k + 1$; AND Prolongation in x-direction;
        Calculate the right hand side;
        Relaxation;
    }
    Step 2: Way in y-direction
        analogously
}
```

Observe that this cycle satisfies the *Restriction-Prolongation-Rule.*

The Q-cycle can be implemented in an efficient way. This means that the number of operations of one Q-cycle is proportional to the number of grid points. Observe that it is enough to find an implementation such that the number of operations of every procedure on the actual level is proportional to the number of grid points of the actual level. Except for the procedure *Calculation of the right hand side*, it is simple to see how to do this.

In case of the discretization with semi-orthogonality, the *Calculation of the right hand side* is similar to the full grid case.

Let us assume that at the beginning of the Q-cycle it is for $(x, y) \in \overset{\circ}{\mathcal{D}}_n$

$$F_{(x,y)} = F_{(x,y)}^{T(x),n-T(x)+1}. \tag{8}$$

Now, we do the *Calculation of the right hand side* in the multilevel cycle in the following way. After a *restriction in x-direction* we use the equation

$$F^{k,l-1}_{(x,y)} = F^{k,l}_{(x,y)} + \frac{1}{2}\left(F^{k,l}_{(x,y-2^{-l})} + F^{k,l}_{(x,y+2^{-l})}\right) - \tilde{a}_h\left(\tilde{u}^{k,l} - \tilde{u}^{k,l-1}, v^{k,l-1}_{(x,y)}\right)$$

in the *Calculation of the right hand side.* After a *prolongation in x-direction* we use the equation

$$F^{k,l}_{(x,y)} = F^{k,l-1}_{(x,y)} - \frac{1}{2}\left(F^{k,l}_{(x,y-2^{-l})} + F^{k,l}_{(x,y+2^{-l})}\right) + \tilde{a}_h\left(\tilde{u}^{k,l} - \tilde{u}^{k,l-1}, v^{k,l-1}_{(x,y)}\right).$$

Similar equations must be used after the *restriction* and *prolongation in y-direction.* At the end of one Q-cycle equation (8) is correct again.

5 Numerical Results

Numerical Example 1 (Spectral Radius of the Q-cycle)

Let $\epsilon > 0$. Then, the bilinear form

$$\begin{aligned} a : \overset{\circ}{W}{}_2^1(\Omega) \times \overset{\circ}{W}{}_2^1(\Omega) &\mapsto \mathbb{R} \\ a(u,v) &= \int_\Omega (\nabla u)^T \begin{pmatrix} \epsilon & \\ & 1 \end{pmatrix} \nabla v \; d(x,y) \end{aligned}$$

is $\overset{\circ}{W}{}_2^1(\Omega)$-elliptic. We are interested in the spectral radius of the Q-cycle iteration matrix on the regular sparse grid $\mathcal{D}_n$. Table 1 shows the approximate spectral radius. It is very small independent of n and ϵ.

ϵ	0.001	0.01	0.1	1	10	100	1000
$n=3$	0.1	0.03	0.02	0.01	0.005	0.02	0.1
$n=4$	0.08	0.002	0.01	0.002	0.002	0.01	0.06
$n=5$	0.01	0.02	0.005	0.002	0.005	0.01	0.01
$n=6$	0.01	0.01	0.005	0.002	0.01	0.005	0.01
$n=7$	0.01	0.01	0.003	0.01	0.01	0.01	0.01
$n=8$	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Table 1: Approximate spectral radius of the Q-cycle

Numerical Example 2 (Convergence of the discretization with semi-orthogonality)

Let us look to the domain

$$\Psi = \left\{(x,y) \in]0,1[^2 | 0 < x < 1 \text{ and } 0.5 \cdot (1 + \sin(\pi \cdot x)) > y > x \cdot 0.25\right\}.$$

n	3	4	5	6	7	8	9
$\|\| - \|\|_{2,\mathcal{D}_n}$	1.5e-3	5.6e-4	1.9e-4	5.8e-5	1.8e-5	5.3e-6	1.6e-6
$\frac{\|\|-\|\|_{2,\mathcal{D}_{n-1}}}{\|\|-\|\|_{2,\mathcal{D}_n}}$	2.0	2.8	3.0	3.2	3.3	3.4	3.4
$\|\| - \|\|_{\infty,\mathcal{D}_n}$	5.4e-3	1.9e-3	6.0e-4	1.9e-4	5.9e-5	1.8e-5	5.2e-6
$\frac{\|\|-\|\|_{\infty,\mathcal{D}_{n-1}}}{\|\|-\|\|_{\infty,\mathcal{D}_n}}$	2.0	2.9	3.1	3.2	3.2	3.3	3.4

Table 2: Convergence of the discretization with semi-orthogonality and $\eta = 1$

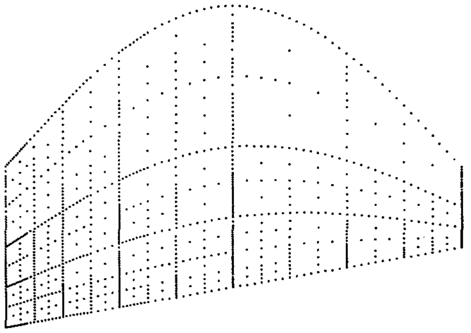


Figure 4. *Adaptive sparse grid on the domain* Ψ *for* $\eta = 0.1$

The function

$$u(x,y) = (1.0 - exp(x/\eta)) \cdot (1.0 - exp(y/\eta))$$

is the solution of the equation $u \in W_2^1(\Psi)$ and

$$a(u,v) = 0 \quad \text{for all } v \in \overset{\circ}{W}{}_2^1(\Psi) \qquad (1)$$

with Dirichlet boundary conditions. Let us map the domain Ψ by a smooth mapping onto the unit square.

This gives a transformed elliptic equation of equation (1) on the unit square. Now, we can solve this equation by the discretization with semi-orthogonality. Thus, we get a discrete solution u_h of the equation (1). Figure 4 shows an adaptive sparse grid with 1220 grid points. There are more points on the left side of the domain, because u is not very smooth for small x.

We use the following discrete norms $||w||_{\infty,\mathcal{D}_n} := \max_{z\in\mathcal{D}_n} |w(z)|$ and $||w||_{2,\mathcal{D}_n} := \sqrt{\frac{\sum_{z\in\mathcal{D}_n} |w(z)|^2}{|\mathcal{D}_n|}}$. Table 2 leads to the conjecture that u_h converges to u with the order

$$||w||_{\infty,\mathcal{D}_n} = O(h^2 \log h^{-1}) \;\; \text{and} ||w||_{2,\mathcal{D}_n} = O(h^2 \log h^{-1}).$$

Acknowledgement. I would like to thank Professor C. Zenger for many helpful discussions.

6 REFERENCES

[1] Bungartz, H.-J., An adaptive Poisson solver using hierarchical bases and sparse grids, in de Groen, P. and Beauwens, R., editors, *in Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra, Brüssel, April, 1991*, Elsevier, Amsterdam, 1992.

[2] Zenger, C., Sparse Grids, in Hackbusch, W., editor, *Parallel Algorithms for Partial Differential Equations: Proceedings of the Sixth GAMM-Seminar, Kiel, January 1990*, volume 31 of *Notes on Numerical Fluid Mechanics*, Vieweg, Braunschweig, 1991.

[3] Balder, R. and Zenger, C., The d-dimensional Helmholtz equation on sparse grids, SFB-Report 342/21/92 A, Technische Universität München, 1992.

[4] Mulder, W., A new multigrid approach to convection problems, *J. Comput. Phys.*, 83:303–323, 1989.

[5] Naik, N. H. and Rosendale, J. V., The improved robustness of multigrid elliptic solvers based on multiple semicoarsening grids, *SIAM J. Numer. Anal.*, 30(1):215–229, 1993.

[6] McCormick, S., *Multilevel Projection Methods for Partial Differential Equations*, volume 62 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM, Philadelphia, 1992.

[7] Pflaum, C., A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids, SFB-Report 342/12/94 A, Technische Universität München, 1994.

[8] Pflaum, C., Anwendung von Mehrgitterverfahren auf dünnen Gittern, 1992.

[9] Pflaum, C. and Rüde, U., Gauss Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids, SFB-Report 342/13/93 A, Technische Universität München, 1993.

[10] Hackbusch, W., *Theorie und Numerik elliptischer Differentialgleichungen*, Teubner, Stuttgart, 1986.

ERROR AND COMPLEXITY ANALYSIS FOR A COLLOCATION-GRID-PROJECTION PLUS PRECORRECTED-FFT ALGORITHM FOR SOLVING POTENTIAL INTEGRAL EQUATIONS WITH LAPLACE OR HELMHOLTZ KERNELS

J. R. Phillips*

Dept. of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139.

SUMMARY

In this paper we derive error bounds for a collocation-grid-projection scheme tuned for use in multilevel methods for solving boundary-element discretizations of potential integral equations. The grid-projection scheme is then combined with a precorrected-FFT style multilevel method for solving potential integral equations with $\frac{1}{r}$ and e^{ikr}/r kernels. A complexity analysis of this combined method is given to show that for homogeneous problems, the method is order $n \log n$ nearly independent of the kernel. In addition, it is shown analytically and experimentally that for an inhomogeneity generated by a very finely discretized surface, the combined method slows to order $n^{4/3}$. Finally, examples are given to show that the collocation-based grid-projection plus precorrected-FFT scheme is competitive with fast-multipole algorithms when considering realistic problems and $1/r$ kernels, but can be used over a range of spatial frequencies with only a small performance penalty.

1. INTRODUCTION

In the last several years, there has been a significant increase in the volume of research on discretized integral equation, or boundary-element, solvers[1]. Boundary-element methods have always been an appealing approach for solving exterior problems, because such methods only discretize domain boundaries and *not* exterior volumes. The main difficulty with boundary-element methods is that they generate dense matrices which were expensive to solve. What has generated renewed interest in boundary-element methods is that the combination of iterative solvers, such as Krylov-subspace methods, and matrix sparsification techniques, like fast-multipole and multilevel methods, have been used to create very fast boundary-element codes [2, 3, 4].

Fast-multipole based codes for solving potential problems with $\frac{1}{r}$ kernels are now commonly used in a variety of engineering applications [5]. What is now of primary research interest is developing sparsification procedures for boundary-element matrices which are capable of solving potential problems with relatively general kernels, at least including $\frac{1}{r}$ and $\frac{e^{ikr}}{r}$ for a wide range of kr [6, 7, 8, 9, 3, 10]. Such a direction parallels the recent work on using multigrid methods to solve the Helmholtz equation [11, 12].

*This work was supported by ARPA contracts N00174-93-C-0035 and J-FBI-92-196 as well as grants from the Consortium for Superconducting Electronics, the Semiconductor Research Corporation (SJ-558), IBM and Digital Equipment Corporation, and an NDSEG fellowship.

In this paper we analyze errors and complexity for a general collocation-grid-projection scheme for use in a precorrected-FFT style algorithm for solving integral equations with general kernels. In the next section, we briefly review the boundary-element method for solving potential integral equations and give a brief description of the precorrected-FFT approach. In Section 3, which contains the main theoretical results of this paper, we give rigorous error bounds for a collocation-based grid-projection scheme. In Section 4, we address the issues of algorithm computational complexity, and analyze the homogeneous case as well as one type of inhomogeneity. In Section 5, we give some experimental results to show that the collocation-based grid-projection plus precorrected-FFT scheme is competitive with fast-multipole algorithms when considering realistic problems and $1/r$ kernels, but can be used over a range of spatial frequencies with only a small performance penalty.

2. PROBLEM FORMULATION AND THE PRECORRECTED-FFT ALGORITHM

Laplace or Helmholtz problems, with a combination of Neumann or Dirichlet boundary conditions, can be cast into an integral equation form using monopole, dipole or combined-layer potentials [13]. In the combined-layer case, the potential is represented by

$$\psi(x) = \int_S \{G_n(x,x') - i\eta G(x,x')\}\sigma(x')da', \qquad x \in S \tag{1}$$

where $x, x' \in \Re^3$, S is a multiply-connected two dimensional surface in $\Re^3$, $G(x,x') = e^{ik||x-x'||}/4\pi||x-x'||$ is the Green's function for the Laplace ($k=0$) or Helmholtz equation, G_n is the surface normal derivative of G at x', $\sigma(x')$ is the combined-layer density often referred to as a charge density, and η is a complex scalar which depends on k.

For each point x for which $u(x)$ is specified, the charge density satifies

$$\frac{\sigma(x)}{2} + \int_S G_n(x,x')\sigma(x')da' - i\eta\int_S G(x,x')\sigma(x')da' = u(x) \tag{2}$$

and for each point x where $u_n(x)$ is specified the charge density satisfies

$$\frac{\partial}{\partial n(x)}\int_S G_{n'}(x,x')\sigma(x')da' + i\eta\frac{\sigma(x)}{2} - i\eta\frac{\partial}{\partial n(x)}\int_S G(x,x')\sigma(x')da' = u_n(x). \tag{3}$$

Boundary-Element Discretization

Boundary-element methods are commonly used to solve potential integral equations like (2) and (3), but are easiest to describe when considering the simple first-kind integral equation of the form

$$\psi(x) = \int_S \sigma(x')G(x,x')da', \qquad x \in S. \tag{4}$$

To compute an approximation to σ, the boundary-element approach is to consider an expansion of the form

$$\sigma(x) \approx \sum_{i=1}^{n} q_i h_i(x), \tag{5}$$

where $h_1(x), ..., h_n(x) : \Re^3 \rightarrow \Re$ are a set of compactly supported expansion functions, and $q_1, ..., q_n$ are the unknown expansion coefficients. The expansion coefficients are then determined by requiring that they satisfy a Galerkin condition of the form

$$Pq = \overline{p}, \tag{6}$$

where $P \in \Re^{n\times n}$ is given by

$$P_{ij} = \int_S h_i(x) \int_S h_j(x')G(x,x')da'da. \tag{7}$$

The approach used in many engineering applications is to approximate the surface S with N planar quadrilateral and/or triangular panels, in which case the support for h_i is just a single panel.

The precorrected-FFT technique

If Gaussian elimination is used to solve (6), $O(n^3)$ operations and $O(n^2)$ storage are required. Typical engineering problems may have thousands or tens of thousands of panels, so that Gaussian elimination is not a feasible approach. In [14, 15] it was shown that the precorrected-FFT method described below is an efficient approach to solving (6), reducing the number of operations and memory required to nearly $O(n \log n)$. As can be seen from Fig. 1, for solution of Laplace's in typical engineering geometries, the precorrected-FFT method is superior to fast multipole algorithms in terms of computation time and memory requirements.

Consider solving (6) by using a Krylov-subspace technique such as GMRES [16]. The dominant costs of such an algorithm are in calculating the n^2 entries of P using (7) before the iterations begin, and performing n^2 operations to compute the dense matrix-vector product on each iteration. To develop a faster approach to computing the matrix-vector product, after discretizing the problem into n panels, consider subdividing the problem domain into an array of small cubes so that each small cube contains only a few panels. Several sparsification techniques for P are based on the idea of directly computing only those portions of Pq associated with interactions between panels in neighboring cubes. The rest of Pq is then somehow approximated to accelerate the computation [2].

One approach to computing distant interactions is to exploit the fact that evaluation points distant from a cube can be accurately computed by representing the given cube's charge distribution using a small number of weighted point charges [17]. Pq can then be approximated in four steps: (1) project the panel charges onto a uniform grid of point charges, (2) compute the grid potentials due to grid charges, (3) interpolate the grid potentials onto the panels, and (4) directly compute nearby interactions. This process is summarized in Figure 2.

Example	Speed	Memory
micromotor	0.68	0.81
cube	0.73	0.31
woven bus	0.63	0.42
bus crossing	0.43	0.26
via	1.42	0.37
DRAM cell	0.80	0.73

Figure 1: Comparison of performance of FFT-based to multipole-based codes for $1/r$ Green function. "Speed" is ratio of matrix-vector product time of precorrected-FFT method to fast multipole based method, "memory" the ratio of required storage.

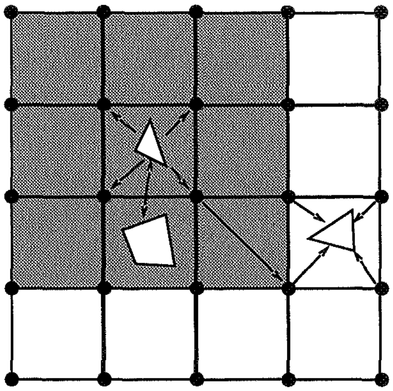


Figure 2: 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid.

There are several possible approaches to computing the grid charge. Analysis of one possible scheme is presented in Section 3. When the grid charges have been determined, their potentials at the grid points must be computed. The potential $\psi(\boldsymbol{x})$ at a point $\boldsymbol{x} = (x, y, z)$ is the sum of the potentials from all the grid charges $q(\boldsymbol{x}')$,

$$\psi(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} g(\boldsymbol{x}, \boldsymbol{x}') q(\boldsymbol{x}') \ . \tag{8}$$

The free-space Green function $g(\boldsymbol{x}, \boldsymbol{x}') = g(x - x', y - y', z - z')$ depends only on the relative difference between the points $\boldsymbol{x}$ and $\boldsymbol{x}'$. Therefore, because of the regular grid, the computation of the grid-charge potentials at the grid points is a three-dimensional discrete convolution. This convolution can be rapidly computed by using the Fast Fourier Transform[18], requiring $O(N \log N)$ operations. Once the grid potentials have been computed, they must be interpolated to the panels.

In the computation of panel potentials due to grid charges, the portions of Pq associated with neighboring cube interactions have already been computed, though this close interaction has been poorly approximated in the projection/interpolation. Before computing a better approximation, it is necessary to remove the contribution of the inaccurate approximation. It is possible to construct a "precorrected" direct interaction operator, $P^{cor}_{a,b}$, which consists of the direct interaction operator $P_{a,b}$ for neighboring cells a and b, with the errors introduced by the grid-charges exactly subtracted out. When used in conjunction with the grid charge representation, $P^{cor}_{a,b}$ results in exact calculation of the interactions between panels which are close. Assuming that the Pq product will be computed many times in the inner loop of an iterative algorithm, $P^{cor}_{a,b}$ will be expensive to initially compute, but will cost no more to subsequently apply than $P_{a,b}$.

3. GRID-PROJECTION SCHEME

In this section, we describe and analyze accurate operators for projecting charge densities onto the grid and for interpolating potentials from the grid, the two problems being equivalent as noted in [3].

The Collocation Grid-projection and Interpolation Operators

Consider approximating the potential of a charge distribution $\rho(x)$ by a set of N_G point charges, $Q_j, j = 1 \dots N_G$ which are positioned at points x_j. Suppose also that both the point charges and the charge distribution lie entirely inside a sphere of radius a centered at the origin. We will require that the potential of the point charges and the potential of the true charge density match at a set of $N \leq N_G$ collocation points $x_{c,k}, k = 1 \dots N$ on a closed surface which encompasses the sphere of radius a. That is, for each k,

$$\sum_j Q_j G(x_j, x_{c,k}) = \int \rho(x') G(x', x_{c,k}) dx'$$

where $G(x, x')$ is the relevant Green's function. It will be convenient if the surface is chosen to be a sphere of radius $r_c > a$, and the collocation points are chosen to be the abscissas of a quadrature rule on the sphere. Integration rules of arbitrary order on a sphere can be constructed by product techniques, but more efficient non-product rules exist [19] which will generally be sufficient for our purposes. By careful selection of the quadrature rule, at least for the orders we have checked, it is possible to insure the grid charge does not substantially exceed the net cube charge. That is, for appropriately selected quadrature rules,

$$\sum_j |Q_j| \leq \kappa \int |\rho(x')| dx' \tag{9}$$

where κ is a constant independent of order.

In addition to constructing operators that represent panel charges by grid charges, it is necessary to construct operators, of comparable accuracy, that interpolate potentials at the grid points to the charge panels.

Lemma 1. If W is an operator which projects charge onto a grid, W^T is an operator which interpolates potential at grid points onto charge coordinates, and W and W^T have comparable accuracy.

Proof. Suppose that a unit charge at the point x_0 is represented by the vector of grid charges q_g. The approximate potential $\hat{\Psi}(y)$ at a point y is given by

$$\hat{\Psi}(y) = \sum_i g(x_i, y) q_g \equiv g^T q_g$$

where x_i is the position of the ith charge, and $g(x_i, y)$ the Green function. Conversely, suppose there is a unit charge at y, and the potential at x_0 is to be computed. Then, if V is the interpolation operator,

$$\hat{\Psi}(x_0) = \sum_i V(x_0, x_i) g(x_i, y) = Vg$$

For a symmetric Green function, $\Psi(x_0) = g(x_0, y) = g(y, x_0) = \Psi(y)$, so that

$$\hat{\Psi}(x_0) - \Psi(x_0) = Vg - \Psi(x_0) = (g^T V^T)^T - \Psi(y) = \hat{\Psi}(y) - \Psi(y) = (g^T q_g) - \Psi(y)$$

if we require $V = q_g^T$. In other words, if W is an operator which represents a charge at point x_0 by grid charges, W^T interpolates potential at the grid points onto the point x_0, and W and W^T have the same order of accuracy. □

Error analysis

First we establish error bounds for the approximation of a panel charge potential by grid charges.

Lemma 2. Suppose a grid-charge representation of a charge distribution $\rho(\boldsymbol{x})$, of total charge $Q = \int |\rho(\boldsymbol{x}')| d\boldsymbol{x}'$, lying inside a sphere $S(a)$ of radius a centered at the origin, has been constructed. Assume the grid charges Q_j are given at points $\boldsymbol{x}_j$, $j = 1 \ldots N_G$, and define $Q_g = \sum_{j=1}^{N} |Q_j|$. The error ϕ_e in the grid-charge approximation of the potential in the $k = 0$ case satisfies

$$|\phi_e| \leq \frac{Q + Q_G}{r_m} (\frac{a}{r_m})^{M+1} \frac{(M+1)^2 + 1}{1 - (a/r_m)} \tag{10}$$

where M is the order of the quadrature rule and r_m is the distance of the nearest potential-evaluation evaluation point to the origin, $r_m > a$.

Proof. The multipole expansion of potential ϕ of the charge distribution is [20]

$$\phi(r, \theta, \phi) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r^{l+1}} [\int_{S(a)} r'^{l} \rho(\boldsymbol{x}') Y_{lm}^*(\theta', \phi') d\boldsymbol{x}'] Y_{lm}(\theta, \phi). \tag{11}$$

Similarly, the multipole expansion of the grid-charge potential $\phi_g(r, \theta, \phi)$ is

$$\phi_g(r, \theta, \phi) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r^{l+1}} [\sum_{j=1}^{N_G} Q_j r_j^l Y_{lm}^*(\theta_j, \phi_j))] Y_{lm}(\theta, \phi). \tag{12}$$

Let (r_c, θ_k, ϕ_k) denote the kth collocation point, $k = 1 \ldots N$, on the surface of the sphere of radius r_c. Assume that the (θ_k, ϕ_k) are the abscissas of a quadrature rule on a sphere such that the rule exactly integrates spherical polynomials of degree at least $2M$. Let $w_k, k = 1 \ldots N$ denote the quadrature rule weights corresponding to a sphere of radius unity.

At a collocation point, the error in the potential, $\phi_e(r_c, \theta_k, \phi_k) = \phi(r_c, \theta_k, \phi_k) - \phi_g(r_c, \theta_k, \phi_k)$ is zero, so we may write

$$\sum_{l=0}^{M} \sum_{m=-l}^{l} \frac{1}{r_c^{l+1}} q_{l,m} Y_{lm}(\theta_k, \phi_k) = \tag{13}$$

$$4\pi \sum_{l=M+1}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r_c^{l+1}} \left[\int_{S(a)} r'^{l} \rho(\boldsymbol{x}') Y_{lm}^*(\theta', \phi') d\boldsymbol{x}' - \sum_{j=1}^{N_G} Q_j r_j^l Y_{lm}^*(\theta_j, \phi_j) \right] Y_{lm}(\theta_k, \phi_k)$$

for $k = 1 \ldots N$, where $q_{l,m}$ is given by

$$q_{l,m} = \left[\frac{4\pi}{2l+1} \int_{S(a)} r'^l \rho(\boldsymbol{x}') Y^*_{lm}(\theta', \phi') d\boldsymbol{x}' - \sum_{j=1}^{N_G} Q_j r_j^l Y^*_{lm}(\theta_j, \phi_j) \right].$$

Multiplying each side of (13) by $w_k Y^*_{l'm'}(\theta_k, \phi_k)$ and summing over k leads to a simplified form. From the identity

$$\int d\Omega \, Y^*_{l'm'}(\theta, \phi) Y_{lm}(\theta, \phi) = \delta_{ll'} \delta_{mm'},$$

and the quadrature rule for selecting w_k, θ_k, ϕ_k, it then follows that

$$\sum_k w_k Y^*_{l'm'}(\theta_k, \phi_k) Y_{lm}(\theta_k, \phi_k) = \delta_{ll'} \delta mm'$$

for $l + l' <= 2M$. Therefore,

$$\frac{1}{r_c^{l'+1}} q_{l',m'} = \tag{14}$$

$$4\pi \sum_{k=1}^{N} w_k \sum_{l=M+1}^{\infty} \sum_{m=-l}^{l} \frac{1}{2l+1} \frac{1}{r_c^{l+1}} \left[\int_{S(a)} r'^l \rho(\boldsymbol{x}') Y^*_{lm}(\theta', \phi') d\boldsymbol{x}' - \sum_{j=1}^{N_G} Q_j r_j^l Y^*_{lm}(\theta_j, \phi_j) \right] Y^*_{l'm'}(\theta_k, \phi_k) Y_{lm}(\theta_k, \phi_k)$$

The addition theorem for spherical harmonics states

$$\frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y^*_{l,m}(\theta', \phi') Y_{l,m}(\theta, \phi) = P_l(\cos\gamma)$$

where γ is the angle between (θ', ϕ') and (θ, ϕ) and $P_l(\cos\gamma)$ a Legendre polynomial. The addition theorem provides a bound

$$\left| \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y^*_{l,m}(\theta_j, \phi_j) Y_{l,m}(\theta_k, \phi_k) \right| \leq 1$$

since $|P_l(\cos\gamma)| \leq 1$, as well as a bound on the magnitudes of the spherical harmonics,

$$|Y^*_{l',m'}(\theta_k, \phi_k)| \leq \sqrt{\frac{2l'+1}{4\pi}}.$$

Since

$$\int_{S(a)} \sum_{m=-l}^{l} \frac{4\pi}{2l+1} r'^l \rho(\boldsymbol{x}') Y^*_{lm}(\theta', \phi') Y_{l,m}(\theta_k, \phi_k) d\boldsymbol{x}' \leq Qa^l$$

and using the additional fact $\sum_k w_k = 4\pi$, we can bound the sum of the infinite series on the right-hand side of (14) to obtain a bound on the (l', m') multipole coefficient of the error

$$|q_{l',m'}| \leq r_c^{l'}(Q+Q_G)(4\pi)\sqrt{(2l'+1)/4\pi}\sum_{l=M+1}^{\infty}(\frac{a}{r_c})^l \tag{15}$$

or

$$|q_{l',m'}| \leq r_c^{l'}(Q+Q_G)(4\pi)\sqrt{(2l'+1)/4\pi}(\frac{a}{r_c})^{M+1}\frac{1}{1-(a/r_c)} \equiv q_{l'}. \tag{16}$$

Using the multipole expansion truncation bound in [2] a bound can be derived for the error in the potential, $|\phi_e|$,

$$|\phi_e| \leq \sum_{l=0}^{M}\frac{q_l}{r^{l+1}}\sqrt{(2l'+1)/4\pi} + \frac{Q+Q_G}{r}(\frac{a}{r})^{M+1}\frac{1}{1-(a/r)}. \tag{17}$$

After substituting the expression for q_l from (16), (17) becomes

$$|\phi_e| \leq \frac{1}{r}(Q+Q_G)(\frac{a}{r_c})^{M+1}\frac{1}{1-(a/r_c)}\sum_{l=0}^{M}(2l+1)(\frac{r_c}{r})^l+\frac{Q+Q_G}{r}(\frac{a}{r})^{M+1}\frac{1}{1-(a/r)}. \tag{18}$$

Depending on the relative size of r_c and r, we may obtain two bounds on the magnitude of the error,

$$|\phi_e| \leq \frac{Q+Q_G}{r}\left[(\frac{a}{r_c})^{M+1}\frac{(M+1)^2}{1-(a/r_c)} + (\frac{a}{r})^{M+1}\frac{1}{1-(a/r)}\right] \qquad r_c < r \tag{19}$$

and

$$|\phi_e| \leq \frac{Q+Q_G}{r}\left[(\frac{a}{r})^{M+1}\frac{(M+1)^2}{1-(a/r_c)} + (\frac{a}{r})^{M+1}\frac{1}{1-(a/r)}\right] \qquad r_c > r. \tag{20}$$

In the potential evaluation process, the worst-case error will occur at the point of smallest r. If we require that $r_c \geq r_m$, the lemma is proved. □

We now have the main result of the paper.

Theorem 1. Suppose the potential of a point charge is given by $1/r$. The grid-based technique for evaluating, outside a sphere of radius r_m, the potential of a charge density of total charge magnitude Q, located inside a sphere of radius a, has error ϕ_e bounded by

$$|\phi_e| \leq (1+\kappa)\frac{Q}{r_m}(\frac{a}{r_m})^{M+1}\frac{(M+1)^2+1}{1-(a/r_m)} \tag{21}$$

where $2M$ is the order of a quadrature rule on a sphere.

Proof. The theorem follows directly from Lemmas 1 and 2. □

Helmholtz Kernels

Suppose that outside a sphere of radius a, a function ψ satisfying the Helmholtz equation is represented by a multipole expansion whose moments up to order N vanish

$$\psi(r,\theta,\phi) = \sum_{l=N+1}^{\infty} \sum_{m=-l}^{l} p_{l,m} h_l^{(1)}(kr) Y_{lm}(\theta,\phi) \tag{22}$$

where k is the wavenumber and $h_l^{(1)} = j_l(kr)+iy_l(kr)$ is the first-kind Hankel function of order l.

For such a potential the following lemma exists [7]:

Lemma 3. For $N > ka$ and any $r > a$, there exists a $c > 0$ such that

$$|\psi(r,\theta,\phi)| \leq c(\frac{a}{r})^{N+1} \tag{23}$$

Theorem 2. Suppose the potential of a charge is given by e^{ikr}/r. If the collocation points in the grid-charge assignment are chosen to be the abscissas of a quadrature rule which exactly integrates spherical harmonics of order $\leq 2ka$, i.e.,

$$M > ka \tag{24}$$

for a quadrature rule of order $2M$, then the grid-based technique for evaluating, outside a sphere of radius r_m, the potential of a charge density of total charge magnitude Q, located inside a sphere of radius a, has error ϕ_e bounded by

$$|\phi_e| \leq c(1+\kappa)\frac{Q}{r_m}(\frac{a}{r_m})^{M+1}\frac{(M+1)^2+1}{1-(a/r_m)} \tag{25}$$

Proof. Given the conditions of Lemma 3, the proof follows exactly as for Theorem 1.

□

Applications and competing approaches

While the grid operators described here were developed with the precorrected-FFT technique in mind, they can be incorporated into any multi-level scheme [10, 3]. The representation described here has two advantages which allow it to be efficient. First, because of the regular spacing of the grid charges, fast ($O(l^2)\log l$, where l is the order of the quadrature rule) translation and potential evaluation operators exist. It appears that in the approach in [10], only the $O(l^4)$ direct operators are available. Secondly, the sharing of grid charges between computational cells allows for a reduction in the total number of coefficients needed to represent the potential in each cell of the computational domain. That is, if there are N cells in the domain, and p^3 grid charges are used to represent the potential in each cell, then, for large N where we may neglect edge effects, the total number of grid charges is only $N(p-1)^3$, a significant reduction for small p. For most engineering problems, we expect $p \leq 5$, so the sharing effect will still be significant. An additional advantage of the grid-based approach is that the potential *throughout* the domain can be obtained at little additional cost once the panel charges have been determined [21].

4. COMPLEXITY ANALYSIS

We first consider the case where the panel charges are evenly distributed throughout space.

Theorem 3. For a homogeneous distribution of N panels, the precorrected-FFT method requires $O(N \log N)$ operations to perform a potential calculation.

Proof. Assume space has been divided into an array of $M \times M \times M$ cells, and that there are about $N = n^3$ panels evenly distributed throughout the $M \times M \times M$ cube, so that there are about $(n/M)^3$ panels in each computational cell. Finally, assume that the grid in each cell is a $p \times p \times p$ array. There are three components in the cost of the precorrected-FFT method. We assume that any costs associated with forming the grid projection operators are negligible, since these calculations only need be performed once, not at each GMRES iteration.

- Cost of direct interactions

$$C_D = \alpha(\frac{n}{M})^6 M^3 = \alpha\frac{n^6}{M^3}$$

- Cost of grid projection and interpolation

$$C_I = \gamma M^3(\frac{n}{M})^3 p^3 = \gamma n^3 p^3$$

  which is independent of M.

- Cost of the FFT

$$C_F = \beta p^3 M^3 \log_2 Mp$$

If we assume that M is proportional to n, then the total cost of the algorithm is $O(n^3 + n^3 \log_2 n) = O(N \log_2 N)$. □

For the boundary-integral methods considered in this paper, however, the panels are usually not homogeneously distributed.

Theorem 4. For a single closed surface at fixed k the precorrected-FFT method requires $O(N^{6/5} \log N)$ operations to perform a potential calculation, where N is the number of panels.

Proof. Again assume space has been divided into an array of $M \times M \times M$ cells, and that the surface measures about n panels wide along each side of the $M \times M \times M$ cube, so that there are about $N \simeq n^2$ panels total, and $(n/M)^2$ panels in each computational cell which is occupied. About M^2 cells will have panels. To determine the complexity of the method, the optimal number of cells M must be determined as a function of problem size, n. The analysis proceeds as above:

- Cost of direct interactions

$$C_D = \alpha(\frac{n}{M})^4 M^2 = \alpha \frac{n^4}{M^2}$$

- Cost of grid projection and interpolation

$$C_I = \gamma M^2 (\frac{n}{M})^2 p^3 = \gamma n^2 p^3$$

  which is independent of M.

- Cost of the FFT

$$C_F = \beta p^3 M^3 \log_2 Mp$$

Neglecting for the purposes of optimization the logarithmic factor, the total cost is

$$C_D = \alpha \frac{n^4}{M^2} + \gamma n^2 p^3 + \beta p^3 M^3$$

which when optimized with respect to M gives

$$M = \left(\frac{2\alpha n^4}{3\beta p^3}\right)^{1/5} \sim n^{4/5}$$

so that

$$C_D \propto n^{12/5} = O(N^{6/5})$$
$$C_I \propto n^2 = O(N)$$
$$C_F \propto n^{12/5} \log_2 np = O(N^{6/5} \log_2 Np)$$

□

In this analysis, we have assumed that p is constant. For a given problem, when solving the Helmholtz discretization as the frequency increases, generally the number of panels must increase to retain a fixed number of panels per wavelength. However, the size of a computational cell decreases proportional to $1/M$, or as $n^{-4/5}$, slower than n. Thus, for high frequencies the criterion in (24) that the order of the quadrature rule be greater than $2k\Delta$ will be violated. We must allow p to vary with n to obtain the correct complexity analysis, which gives a different complexity bound.

Theorem 5. For a single closed surface the precorrected-FFT method with $\sqrt{N}$ proportional to k requires at most $O(N^{4/3} \log N)$ operations to perform a potential calculation, where N is the number of panels.

Proof. Assume the size R of the computational domain is fixed. Further, assume a fixed number of panels per wavelength, $n \sim 1/\lambda \sim k$ is required to maintain the solution accuracy. Then $k\Delta = kR/M \sim n/M$. The number of collocation points necessary for order l quadrature is $O(l^2)$, which is of the same order as the number of grid charges per cell, p^3. Thus we have

$$p \sim (\frac{n}{M})^{2/3}$$

Repeating the above complexity analysis, we have

- Direct cost $C_D = O(n^4/M^2)$
- Interpolation cost $C_I = O(p^3n^2) = O(n^4/M^2)$, same order as the direct cost
- FFT cost $C_F = O(M^3p^3 \log_2 Mp) = O(Mn^2)$

The total cost is thus $C_T = O(Mn^2 + n^4/M^2)$ which when optimized for M gives

$$M = O(n^{2/3})$$

The asymptotic cost of the entire algorithm is then $O(N^{4/3} \log_2 N)$, a slight increase over the $O(N^{6/5})$ in the case of Poisson's equation, and competitive with two-level multipole based schemes for the Helmholtz equation [7].

We should also note that the cost of forming the grid projection operators, $O(p^9) = O(n^2) = O(N)$ remains reasonable. □

5. COMPUTATIONAL RESULTS

Empirical Grid Error Analysis

In Figures 3(a) and 3(b), the errors in the potential due to the grid charge approximation are shown for two values of the collocation sphere radius r_c, in the Laplace ($k = 0$) limit. In Figure 3(a), with r_c small, for all orders of approximation the error decays slowly away from the charge distribution. Since in this case $r_c \simeq r_{min}$, we expect the error to behave essentially as a monopole, dying slowly away from the origin, regardless of the order of the quadrature rule. We only expect the order of quadrature rule to change the constant factor in front of the error term. Notice in Figure 3(b), where r_c is considerably larger, the worst-case errors have not changed much, as predicted by our previous analysis. The variation of error with distance, however, changes drastically. As the collocation sphere radius is increased, the magnitude of the low order multipole coefficients of the error decreases, and the errors decay rapidly with distance. Note that the sharp error decay associated with high order multipole approximation ends at about the collocation sphere radius.

In Figures 3(c) and 3(d), we consider errors in the Helmholtz equation. At low k, all three order schemes considered still exhibit acceptable error properties (if an acceptable worst-case error is of order $10^{-4} - 10^{-3}$). As k is increased, however, the

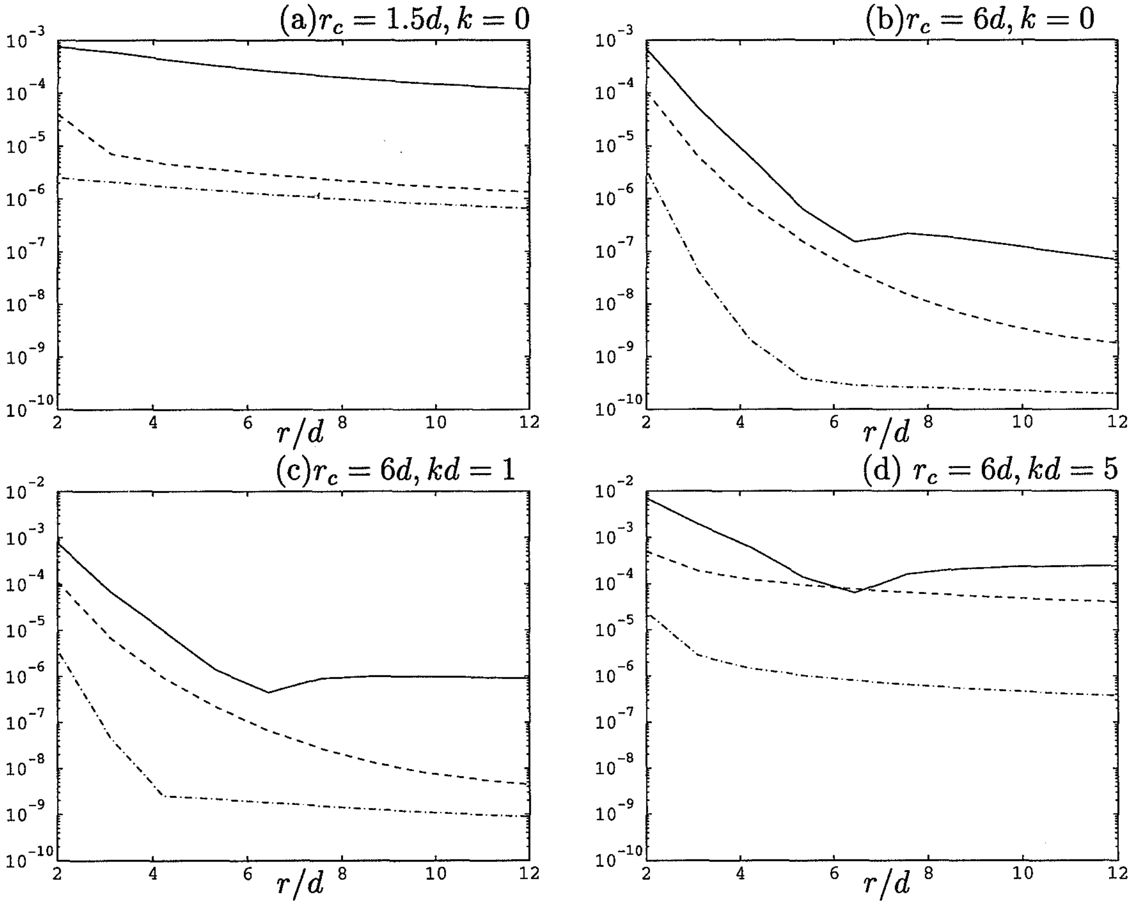


Figure 3: Error in grid approximation of potential of 100 charges of random strength $Q \in [0,1]$ located at random positions inside a cube of side length $2d$ centered at the origin. Collocation sphere radius is $r_c = 1.5d$ (left figure), $r_c = 6d$ (right figure). Solid line: $p = 3$, order 7 quadrature rule. Dash line: $p = 4$, order 11 quadrature rule. Dash-dotted line: $p = 5$, order 14 quadrature rule.

low-order schemes become inaccurate, and the high-order scheme ($p = 5$) becomes less accurate, though still retains acceptable accuracy for this relatively high frequency (at this freqency, the basic computational cell is more than a wavelength long).

Computational Examples

First we analyze the behavior of the precorrected-FFT method as a function of problem size, for Laplace and Helmholtz kernels. A cube is discretized into quadrilateral panels, with n panels along each size. The time required to perform a matrix-vector product, and the memory necessary for the linear system solution, is then tabulated for n ranging from 15 to 100. For the Helmholtz problem, we will require that the discretization have 15 panels per wavelength along each side of the cube. For a unit cell of length Δ, the order p of the grid representation and order M of the quadrature rule are chosen by the rules: $k\Delta \leq 1.75$ corresponds to $p = 3, M = 7$

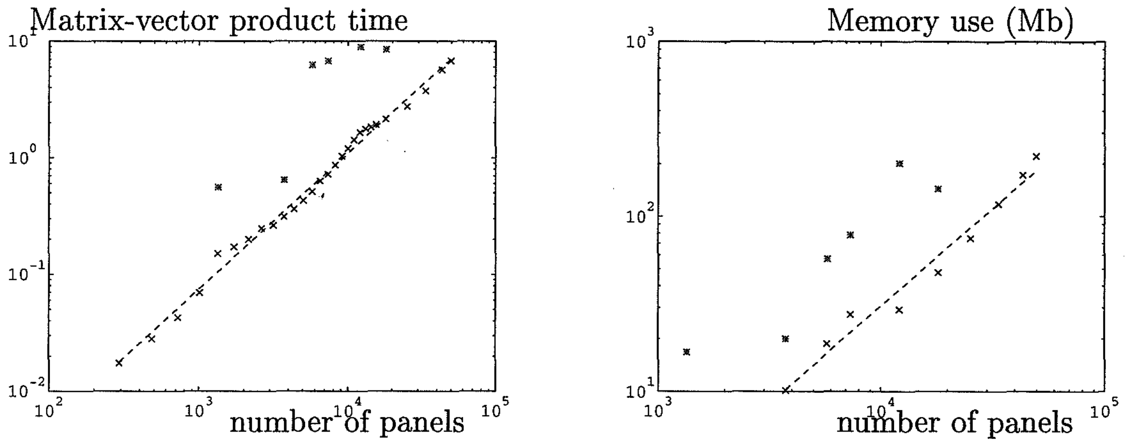


Figure 4: CPU time and memory use for discretized cube. x: Laplace problems. $*$: Helmholtz problems, with $kn = 15$, n the number of panels along a side of the cube. Dash line: best fit line to Laplace data: assumed time, memory $= Cn^\alpha$, computed $\alpha = 1.16$ for CPU time, $\alpha = 1.11$ for memory use.

(26-point rule); $1.75 < k\Delta \leq 2.75$ corresponds to $P = 4, M = 11$ (56-point rule); $k\Delta > 2.75$ corresponds to $p = 5, M = 14$ (72-point rule). The results are shown in Fig. 4.

The results for Laplace's equation follow the expected $O(N^{1.2})$ behavior very closely. Some degree of irregular growth is apparent in the plot as a result of changing grid levels. The cost of the precorrected-FFT method is generally greater when the Helmholtz kernel is used, in part because complex quantities must be manipulated, but mostly because a higher-order grid representation is necessary to accurately represent the charge in a cell. For the range of frequencies considered, the problems with a Helmholtz kernel appear to be roughly a factor of $2 - 10$ slower than the problems with a Laplace kernel. The growth with problem size of computation time and memory usage seems to be fairly irregular, for the choice of grids considered here. The observed irregularity occurs because the order of the approximation must change to maintain a fixed relationship between the wavelength and the size of a computational cell.

Now we demonstrate that the precorrected-FFT technique can accurately compute solutions of integral equations with an oscillatory kernel. Assume a sphere of radius a, with the boundary conditions

$$u(x) = h_3^{(1)}(ka)\sin^2\theta\cos\theta\cos 2\phi$$

which has solution $\psi(r,\theta,\phi) = h_3^{(1)}(kr)\sin^2\theta\cos\theta\cos 2\phi$. The sphere was discretized along longitudes and latitudes, with 50 divisions in each variable, to generate a problem with 2600 panels. We take $k = 4\pi$, corresponding to a sphere 4 wavelengths in diameter. Fig. 5 shows the computed results. The agreement is excellent, and closer inspection shows the error in the computed fields to be less than 10^{-3}, on the order of the GMRES tolerance. We have encountered no computational difficulties at much smaller or moderately larger wavelengths.

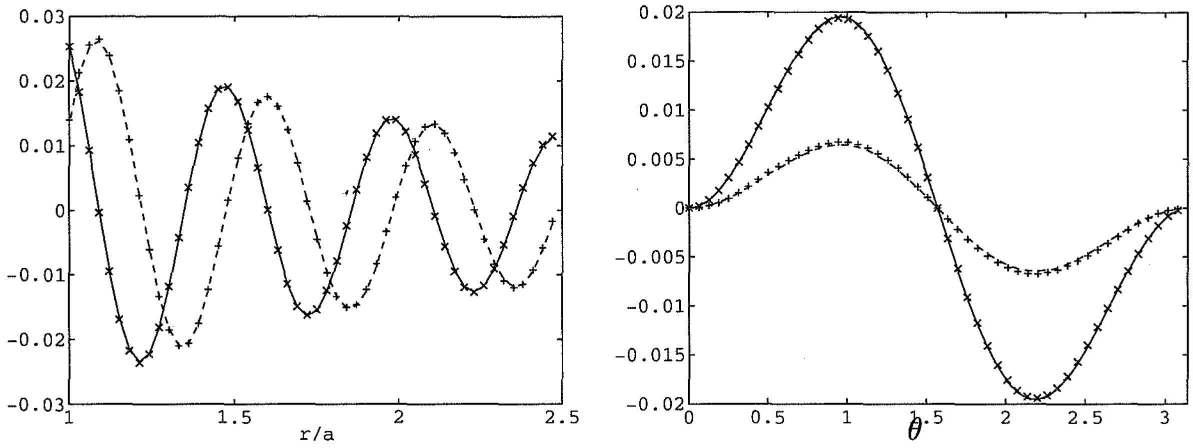


Figure 5: Solid line: real part of exact solution. Dashed line: imaginary part of exact solution. x: computed real part of solution. +: computed imaginary part of solution.

6. CONCLUSIONS

In this paper we described and carefully analyzed a collocation-grid-projection plus precorrected-FFT method for solving potential integral equations with $\frac{1}{r}$ and e^{ikr}/r kernels for a wide range of k. We demonstrated experimentally and analytically that the errors are well-controlled, and showed that the method is competitive with fast-multipole algorithms for $\frac{1}{r}$ kernels but is much more general. It should be noted that the collocation-grid-projection plus precorrected-FFT method can be combined with the multilevel methods in [3] to minimize the effects of inhomogeneity, but we have yet to see the need for such an approach in practical applications.

REFERENCES

[1] R. F. Harrington, *Field Computation by Moment Methods.* New York: MacMillan, 1968.

[2] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems.* Cambridge, Massachusetts: M.I.T. Press, 1988.

[3] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernels," *Computer Physics Communications*, no. 65, pp. 24–38, 1991.

[4] K. Nabors and J. White, "FastCap: A Multipole-Accelerated 3-D Capacitance Extraction Program," *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 10, November 1991, pp. 1447-1459.

[5] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, "Multipole Accelerated Preconditioned Iterative Methods for Three-Dimensional Potential Integral Equations of the First Kind," *SIAM J. on Sci. and Stat. Comp.*, May 1994, Vol. 15, No. 3, pp. 713-735.

[6] V. Rokhlin, "Rapid Solution of Integral Equations of Scattering Theory in Two Dimensions," Journal of Comp. Physics, no. 86, 1990, pp. 414-439.

[7] V. Rokhlin, "Diagonal forms of translation operators for the Helmholtz equation in three dimensions", *Applied and Computational Harmonic Analysis*, vol. 1, pp.82-93, 1993.

[8] F. X. Canning, "Transformations that Produce a Sparse Moment Matrix", Journal of Electromagnetic Waves and Applications, no. 4, 1990, pp. 893-913.

[9] C.C. Lu and W. C. Chew, "Fast Algorithms for solving Hybrid Integral Equations," IEEE Proceedings-H Vol. 140, No. 6, December 1993, pp. 455-460.

[10] C. R. Anderson, "An implementation of the Fast Multipole Method without Multipoles," *SIAM Journal of Sci. Comp.*, vol. 13, No. 4, July 1992, pp. 923-947.

[11] K. Brackenridge, "Multigrid and Cyclic Reduction Applied to the Helmholtz Equation," *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, April 1993, Colorado, pp. 31-42.

[12] B. Engquist and E. Luo, "Multigrid Methods for Differential Equations with Highly Oscillatory Coefficients," *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, April 1993, Colorado, pp. 31-42.

[13] D. Colton and R. Kress, *Integral Equations Methods in Scattering Theory*, Krieger Publishing Company, Malabar, Florida, 1992.

[14] J. White, J. R. Phillips and T. Korsmeyer, "Comparing Precorrected-FFT and Fast Multipole Algorithms for Solving Three-Dimensional Potential Integral Equations," *Proceedings of the Colorado Conference on Iterative Methods*, Breckenridge, Colorado, April 1994.

[15] J. Phillips and J. White, "A Precorrected-FFT method for Capacitance Extraction of Complicated 3-D Structures," *Proceedings of the Int. Conf. on Computer-Aided Design*, Santa Clara, California, November 1994.

[16] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.

[17] L. Berman, "Grid-multipole calculations," Tech. Rep. RC 19068(83210), IBM Research Report, 1993.

[18] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, S.I.A.M. Press, Philadelphia, 1992.

[19] A. D. McLaren, "Optimal numerical integration on a sphere," *Math. Comput.*, vol. 17, 1963, pp. 361-383.

[20] J. D. Jackson, *Classical Electrodynamics*, John Wiley, New York, 1975.

[21] A. Mayo, "The fast solution of Poisson's and the biharmonic equations on irregular regions," *SIAM J. Numer. Anal.*, vol. 21, No. 2, April, 1984, pp. 285-299.

MULTIGRID TECHNIQUES FOR HIGHLY INDEFINITE EQUATIONS

Yair Shapira
Computer Science Department, Technion –
Israel Institute of Technology, Haifa 32000, Israel.

SUMMARY

A multigrid method for the solution of finite difference approximations of elliptic PDEs is introduced. A parallelizable version of it, suitable for two and multi level analysis, is also defined, and serves as a theoretical tool for deriving a suitable implementation for the main version. For indefinite Helmholtz equations, this analysis provides a suitable mesh size for the coarsest grid used. Numerical experiments show that the method is applicable to diffusion equations with discontinuous coefficients and highly indefinite Helmholtz equations.

1 INTRODUCTION

The multigrid method is a powerful tool for the numerical solution of elliptic PDEs [4]. Its rate of convergence, however, deteriorates when non-elliptic problems are encountered; this phenomenon is due to error components (modes, eigenvectors) which have nearly zero eigenvalues with respect to the coefficient matrix. For convection problems, for example, error modes which are smooth in the convection direction are nearly singular and require a special treatment [6] [7]. For indefinite equations, we distinguish two classes of problems: (a) slightly indefinite problems, for which very few modes with negative eigenvalues (say two or three) exist, and (b) highly indefinite problems, for which many more such modes exist. For class (a), the method of [5], which is based on filtering nearly singular modes, achieves convergence rates which are close to those for the Poisson equation. The Cyclic Reduction Multigrid (CR-MG) of [8] is also superior to standard multigrid. For class (b), a projection method (suitable for finite element schemes) is presented in [3]. The AutoMUG method of [16] [17] [18] and a variant of Black Box Multigrid [15] also achieve satisfactory convergence rates especially when supplemented with an acceleration scheme. The two latter methods can also handle diffusion problems with discontinuous coefficients.

The aim of this work is to supply a suitable implementation for AutoMUG for highly indefinite Helmholtz equations. To this end, we introduce a parallelizable version of AutoMUG, called Parallelizable AutoMUG (PAMUG). This method may be considered a generalization of the Parallelizable Superconvergent Multigrid (PSMG) of [11] to nonsymmetric and indefinite problems. PAMUG uses the fine grid at all levels, hence is suitable for parallel architectures with a large number of processors; however, we do not use it as a solver but only as a theoretical tool supplying a suitable implementation for AutoMUG. Due to its simple algebraic formulation, PAMUG is suitable for two-level analysis in some cases. Furthermore, in some model cases, including indefinite Helmholtz equations, the spectrum of the multi level iteration matrix is computable. This enables one to choose in advance a suitable mesh-size for the coarsest grid and a suitable acceleration scheme (if needed). Due to the similarity of AutoMUG and PAMUG, this implementation applies also to AutoMUG, as follows from numerical experiments.

The content of this paper is as follows. In Section 2 AutoMUG and PAMUG are defined. In Section 3 they are analyzed. In Section 4 numerical experiments (using AutoMUG) are reported.

2 THE AutoMUG AND PAMUG METHODS

2.1 Abstract Definition of a Multi Level Method

We start with an abstract definition of a multi level (ML) method for the solution of the linear system of equations

$$Ax = b.$$

In the following, $\tilde{S} : x \to \tilde{S}x$ is a smoothing procedure and ϵ, r, t and o are nonnegative integers denoting, respectively, the cycle index, the number of presmoothings, the number of postsmoothings and the minimal bandwidth of A (with some ordering of variables) for which ML is called recursively. The operators R (restriction), P (prolongation) and Q (coarse grid coefficient matrix) will be defined later.

```
ML($x_{in}, A, b, x_{out}$) :
  if $A$ is of bandwidth $< o$
  for some variable ordering
      $x_{out} \leftarrow A^{-1}b$
  otherwise:
      $x_{in} \leftarrow \tilde{S}x_{in}$    (repeat $r$ times).
      $e \leftarrow 0$                                                (1)
      ML($e, Q, R(Ax_{in} - b), e_{out}$) }
      $e \leftarrow e_{out}$                    } repeat $\epsilon$ times
      $x_{out} \leftarrow x_{in} - Pe$
      $x_{out} \leftarrow \tilde{S}x_{out}$    (repeat $t$ times).
```

An iterative application of ML is given by

```
$x_0 = 0, \; k = 0$
while $||Ax_k - b||_2 \geq \text{threshold} \cdot ||Ax_0 - b||_2$
    ML($x_k, A, b, x_{k+1}$)                                 (2)
    $k \leftarrow k + 1$
endwhile.
```

Below we define the operators R, P and Q of (1) for AutoMUG, its variant AutoMUG(q) and the parallelizable versions PAMUG and PAMUG(q).

2.2 Some Matrix Functions

Let K be a positive integer and I the identity matrix of order K. For any matrix M, $M = (m_{i,j})_{1\leq i,j\leq K}$, define the matrix functions

$$\begin{aligned}
rowsum(M) &= diag(\sum_{j=1}^{K} m_{i,j})_{1\leq i\leq K}\\
D(M) &= diag(M)\\
R(M) &= 2I - MD(M)^{-1}\\
Q(M) &= R(M)M\\
P(M) &= 2I - D(M)^{-1}M\\
S(M) &= rowsum(P(M)).
\end{aligned}$$

These definitions apply to AutoMUG and PAMUG. For AutoMUG(q) and PAMUG(q), replace the above definition of $S(M)$ by $S(M) \equiv (2+q)I$ (the role of the parameter q will be explained later). Let V_K be the space of the $K \times K$-grid functions (it is assumed hereafter that the first point in a grid is numbered $(1,1)$). Define the orthogonal projection $O : V_K \to V_{\lfloor K/2 \rfloor}$ by $(Ov)_{i,j} = v_{2i,2j}$ and the permutation U by

$$(Uv)_{i,j} = v_{j,i}, \quad v \in V_K.$$

For any matrix B, we say that B is a K-block matrix if B is block diagonal with tridiagonal blocks of order K, that is,

$$B = blockdiag(B^{(j)})_{1\leq j\leq K},$$

with

$$B^{(j)} = tridiag(b_i^{(j)}, c_i^{(j)}, d_i^{(j)})_{1\leq i\leq K}, \quad 1 \leq j \leq K.$$

By the notation '*tridiag*' we mean a periodically extended tridiagonal matrix, that is, $b_1^{(j)} = B_{1,K}^{(j)}$ and $d_K^{(j)} = B_{K,1}^{(j)}$. We assume that either

$$b_1^{(j)} = d_K^{(j)} = 0, \quad 1 \leq j \leq K$$

or $K = 2^k$ for some positive integer k. This guarantees that A and the coarse grid coefficient matrices defined bellow are of property-A. Actually, the block submatrices $B^{(j)}$ need not be of the same size; for simplicity, however, we assume that they are. Non-rectangular grids can be embedded into rectangular ones (see [9] [18]).

2.3 Transfer and Coarse Grid Operators

Here we define the operators R, P and Q used in (1) for linear systems which arise, for example, from finite difference approximations of elliptic PDEs.

Let N and n be positive integers, where $n \leq \lfloor \log_2 N \rfloor$ denotes the number of levels minus 1. Assume that A is of the form

$$A = X + Y, \tag{3}$$

where X and UYU are N-block matrices. For example, if

$$X = UYU = blockdiag[tridiag(-1, 2 - \frac{\beta h^2}{2}, -1)], \tag{4}$$

(where β is a parameter and h is the cell size) then A represents a five-point second order discretization of the Helmholtz equation

$$-u_{xx} - u_{yy} - \beta u = f \tag{5}$$

in a square (the unit square is used here).

Define $X_0 = X$ and $Y_0 = Y$. For $i = 1, \ldots, n$, define the matrices R_i, P_i and A_i, in this order, by

$$\begin{aligned} X_i &= S(Y_{i-1})Q(X_{i-1}) \\ Y_i &= S(X_{i-1})Q(Y_{i-1}) \\ R_i &= OR(Y_{i-1})R(X_{i-1}) \\ P_i &= P(X_{i-1})P(Y_{i-1})O^T \\ A_i &= O\,(X_i + Y_i)\,O^T. \end{aligned}$$

These definitions apply to AutoMUG and AutoMUG(q). For the parallelizable versions PAMUG and PAMUG(q), they are modified as follows: omit the operators O and O^T in the above definitions and replace the definition of P_i by $P_i \equiv I$. The parameter q in AutoMUG(q) and PAMUG(q) is chosen by the user such that $S(X_{i-1})$ and $S(Y_{i-1})$ are optimally approximated, in some sense, by $(2+q)I$; for example, if β in (5) varies with the spatial coordinates, then a reasonable choice for q is an average value of $-\beta h^2/4$. PAMUG(q) and AutoMUG(q) are suitable for two-level analysis. For simplicity, $q = 0$ is used in most of this analysis.

The ML procedure, namely ML(x_{in}, A, b, x_{out}) defined in (1), is called $n + 1$ times per iteration. In the $(n + 1)$st time, it is a direct solver. In order to implement AutoMUG, AutoMUG(q), PAMUG or PAMUG(q), the ith call to the ML procedure, $1 \leq i \leq n$, uses the operators

$$Q \leftarrow A_i,\ R \leftarrow R_i \text{ and } P \leftarrow P_i.$$

Note that, for PAMUG and PAMUG(q), A_1 includes four independent subsystems, each of which corresponds to odd (even) numbered variables in the x and y spatial directions (see [18]). Furthermore, the coarse grid equations in PAMUG and PAMUG(q) corresponding to even numbered variables in both spatial directions are identical to those of AutoMUG and AutoMUG(q), respectively. Roughly speaking, these methods have a similar effect on low frequency error components, hence it is likely that convergence rate estimates for the parallelizable versions are fair approximations to those for the sequential ones. This is verified in Corollary 1 and the numerical experiments in Section 4. For certain examples, e.g., convection-diffusion equations with periodic boundary conditions, AutoMUG and PAMUG are equivalent to AutoMUG(0) and PAMUG(0), respectively, because all the row-sums used in AutoMUG and PAMUG are equal to the constant number 2 (as a matter of fact, AutoMUG is equivalent to AutoMUG(0) also for other types of boundary conditions, provided that N is odd). This is also the case for either definite or indefinite Helmholtz equations, provided that an appropriate $q \neq 0$ is used. Hence, one can learn about the features of AutoMUG (which is actually used in our applications) from the analysis of PAMUG, PAMUG(q) and AutoMUG(q).

3 ANALYSIS OF PAMUG AND AUTOMUG

3.1 Two-Level Analysis

Here we derive upper bounds for convergence rates for PAMUG(0) and AutoMUG(0) applied to a class of equations, including Symmetric Positive Definite (SPD) Helmholtz equations (e.g., $\beta h^2/2 < 4\sin^2(\pi h/2)$ in (4)). These bounds are independent of the size of the problem and the clustering of the eigenvalues near zero. This implies that AutoMUG is capable of handling nearly singular eigenvalues; hence, it may solve highly indefinite problems, provided that the negative eigenvalues are handled by a suitable acceleration scheme (see also Section 3.3).

Since PAMUG is designed for parallel implementations, it may be assumed that the damped Jacobi iteration, which is perfectly parallelizable, is used as a smoothing procedure (for some architectures, two damped Jacobi relaxations are less expensive than one red-black Gauss-Seidel sweep). This simplifies the analysis considerably.

The order in which smoothing and coarse grid correcting are performed is immaterial, due to the commutativity of the smoothing and coarse-grid correcting operators. For consistency, however, we consider damped Jacobi iterations for presmoothing and other methods (e.g., Jacobi) for postsmoothing.

Theorem 1 *Assume that*

- *X and Y commute with each other.*
- *$D(X) = D(Y) = I$ (isotropy assumption).*
- *the spectra of X and Y lie in the interval $(0, 2)$ (e.g., X and Y are symmetric M-matrices or symmetric irreducibly diagonally dominant matrices, see [20]).*

Then the convergence factor for a two-level implementation of PAMUG(0) with r damped Jacobi presmoothings (with damping factor $2/3$) and no postsmoothings is bounded from above by

$$\max\left\{3^{-r}, \frac{3r^r}{4(r+1)^{r+1}}\right\} \sim_{r\to\infty} \frac{3}{4e}\cdot\frac{1}{r} \qquad (6)$$

For the proof, see Appendix A.

Corollary 1 *Assume that A is normal. Then Theorem 1, with the bound in (6) multiplied by 2, applies also to AutoMUG(0), provided that one additional postsmoothing of the form $x \leftarrow POx$ is performed.*

For the proof, see Appendix B.

3.2 Multi-Level Analysis for PAMUG

Theorem 1 yields convergence rates for the two-level implementation of PAMUG(0) to essentially semi positive definite problems. This implies that indefinite problems may also be solved, provided that the negative eigenvalues are handled efficiently by an acceleration scheme. In this section, we give quantitative support for this heuristic.

Theorem 2 *Assume that the blocks in X and UYU are circulant Toeplitz matrices, that is,*

$$\begin{aligned} X &= blockdiag[tridiag(b_0, c_0, d_0)] \\ UYU &= blockdiag[tridiag(\beta_0, \gamma_0, \delta_0)] \end{aligned}$$

for some constants b_0, c_0, d_0, β_0, γ_0 and δ_0. Let

$$p_0 = \frac{b_0 + c_0 + d_0}{c_0}, \quad q_0 = \frac{\beta_0 + \gamma_0 + \delta_0}{\gamma_0}.$$

For $0 \le i < n-1$, define

$$\begin{array}{ll} b_{i+1} = -(2-q_i)b_i^2/c_i & c_{i+1} = (2-q_i)(c_i - 2b_i d_i/c_i) \\ d_{i+1} = -(2-q_i)d_i^2/c_i & p_{i+1} = (b_{i+1} + c_{i+1} + d_{i+1})/c_{i+1} \\ \beta_{i+1} = -(2-p_i)\beta_i^2/\gamma_i & \gamma_{i+1} = (2-p_i)(\gamma_i - 2\beta_i\delta_i/\gamma_i) \\ \delta_{i+1} = -(2-p_i)\delta_i^2/\gamma_i & q_{i+1} = (\beta_{i+1} + \gamma_{i+1} + \delta_{i+1})/\gamma_{i+1}. \end{array}$$

Define

$$\begin{aligned} g(c,\gamma;p,q;x,y) &= 1 - \frac{(2-x/c)(2-y/\gamma)(x+y)}{(2-q)x(2-x/c) + (2-p)y(2-y/\gamma)} \\ f_r(c,\gamma;p,q;x,y) &= g(c,\gamma;p,q;x,y)\left(1 - \frac{x+y}{\alpha(c+\gamma)}\right)^r \\ f_r^{(n-1)}(x,y) &= f_r(c_{n-1},\gamma_{n-1};p_{n-1},q_{n-1};x,y). \end{aligned}$$

For $i = n-2, n-3, \ldots, 0$, define

$$\begin{aligned} f_r^{(i)}(x,y) &= f_r(c_i,\gamma_i;p_i,q_i;x,y) \\ &+ f_r^{(i+1)\epsilon}((2-q_i)x(2-x/c_i), (2-p_i)y(2-y/\gamma_i)) \\ &\cdot (1 - g(c_i,\gamma_i;p_i,q_i;x,y))\left(1 - \frac{x+y}{\alpha(c_i+\gamma_i)}\right)^r. \end{aligned}$$

Then there exists an orthogonal matrix T such that the iteration matrix of PAMUG (implemented with cycle index ϵ and r damped Jacobi smoothings with damping factor α^{-1}) is given by

$$T^* diag\{f_r^{(0)}(x,y)\}_{(x,y)\in spect(X)\times spect(Y)}T.$$

For the proof, see Appendix C. Theorem 2 yields an efficient way to compute in advance the spectrum of the iteration matrix of PAMUG. This method is employed below for our model problem.

3.3 The Indefinite Helmholtz Equation

As discussed in [5], the most problematic eigenvalues of indefinite equations are those which are close to zero. Theorem 1 and Corollary 1 show that PAMUG(0) and AutoMUG(0) handle positive eigenvalues arbitrarily close to zero, giving convergence factors which are independent of the size of the problem and the clustering of the eigenvalues. Although this applies to the two-level method and definite problems, it indicates that the algorithm may also be efficient for the multi level method and indefinite problems. In this case, however, the cell-size of the coarsest grid cannot be arbitrarily large, as is shown below.

When the coarsest grid is not too coarse, numerical computations using Theorem 2 show that the PAMUG iteration matrix has only a few eigenvalues of magnitude larger than one. These eigenvalues may be annihilated (their corresponding error components are significantly reduced) by an appropriate Krylov space acceleration method applied to the basic multi level iteration (2). The remaining eigenvalues are considerably smaller (in magnitude) than one; good convergence rates are thus achievable, provided that the dimension of the Krylov space is large enough, say twice as large as the number of eigenvalues of magnitude greater than one. When the number of levels is large, so that very coarse grids are used, the spectrum of the iteration matrix significantly deteriorates; the magnitude of many eigenvalues then approaches one and exceeds it.

Thus, Theorem 2 may help in choosing in advance an appropriate dimension for the Krylov space in the acceleration method. For highly indefinite problems, however, this dimension must be rather large; in this case, a conventional acceleration method, such as GMRES of [14], will not do, since the required amount of storage (respectively, arithmetical operations) increases linearly (respectively, quadratically) with the dimension of the Krylov space used. The Transpose Free Quasi Minimal Residual method of [12] and the Conjugate Gradient Squared method of [19], which use arbitrarily large Krylov spaces with fixed requirements of work and storage, are thus preferable.

Consider the indefinite Helmholtz equation (5) in the unit square with periodic boundary conditions, discretized as in (3), (4). Our aim is to compute the spectrum of the PAMUG iteration matrix for this problem. In this case,

$$spect(X) = spect(Y) = \{4\sin^2(\pi j/N) - \beta/(2N^2)\}_{1\le j<N}.$$

Modes which are constant in either one of the spatial directions are excluded; this is equivalent to assuming that the right hand side includes no Fourier modes which are constant in one of the spatial directions, and the equation is projected onto the linear subspace orthogonal to the set of these modes. This situation simulates problems with Dirichlet boundary conditions, since the spectrum of X and Y is not enlarged by the transformation

$$\begin{array}{ccc} \text{periodic boundary conditions} & \to & \text{Dirichlet boundary conditions} \\ N & \to & N/2-1 \\ \beta & \to & \beta/4. \end{array}$$

One damped Jacobi smoothing (with damping factor 1/2) and two Jacobi smoothings are used in each level of a V-cycle. This implementation is chosen in order to cancel possible poles of the function g of Theorem 2 (and the proof of Theorem 1) and guarantee that the functions $f^{(i)}$ there are bounded. Indeed, it is verified that no pole of the functions $f^{(i)}$ is encountered during the computation. This choice was the most efficient one; using, e.g., damping factor 1/2 for all the three relaxations yields worse results. This is another place where the theory helps in choosing a suitable implementation; however, it is suitable only for ideal parallel machines, whereas in practice (Section 4) we use AutoMUG with the more efficient red-black Gauss-Seidel relaxation.

The results are displayed in Figures 1 and 2. The last rows of these figures show how the spectrum deteriorates when the coarsest grid is too coarse. Here $\beta = 3200$ and we find that for $N = 256$ and 512, respectively, using 3 and 4 levels yields only a few large eigenvalues. The remaining eigenvalues are contained in $[-0.25, 0.25]$, which implies that the effective rate of convergence should be around 0.25, provided that the large eigenvalues can be handled by the acceleration. Consequently, a 64×64 coarsest grid is suitable for achieving this rate of convergence. In light of the above discussion, it is expected that for Dirichlet problems and $\beta = 800$ the choices $N = 127$ and $N = 255$ yield pictures which are much the same as those of Figures 1 and 2, respectively; hence a 31×31 coarsest grid is suitable in this case. When a further coarser grid, namely, a 15×15 grid, is used, the eigenvalues of the iteration matrix are clustered around ± 0.7; thus, a convergence factor of at least 0.7 is expected in this case (see Table 2 below). It can also be inferred from the figures that the number of levels is immaterial; what matters is the cell-size of the coarsest grid alone. This is in agreement with a result of [3] (see also Table 1 below).

There is also a physical explanation for the above lower bound on the resolution of the coarsest grid. For Equation (5), consider waves of wave number (k, l) satisfying $\pi^2(k^2+l^2) \approx \beta$. Evidently, these waves appear in the solution, since they are amplified by the inverse of the operator. Hence, an appropriate coarse grid must be capable of approximating these modes. In particular, it should be sufficiently fine to approximate the above modes with $k = 0$ (resp., $k = 1$) and $l = 0$ (resp., $l = 1$) for periodic (resp., Dirichlet) boundary conditions. In light of the Nyquist rate, a proper approximation requires 2 points per wave length; this yields roughly $\lfloor N/2^n \rfloor \geq 2\sqrt{\beta}/\pi$.

Another explanation for the above restriction arises from matrix theory. It was observed that for sufficiently fine grids, the coefficient matrix is an L-matrix, that is, has positive main diagonal elements and nonpositive off-diagonal elements. For too coarse grids the amount of indefiniteness is so large that the main diagonal elements become negative, which leads to an inappropriate PDE approximation.

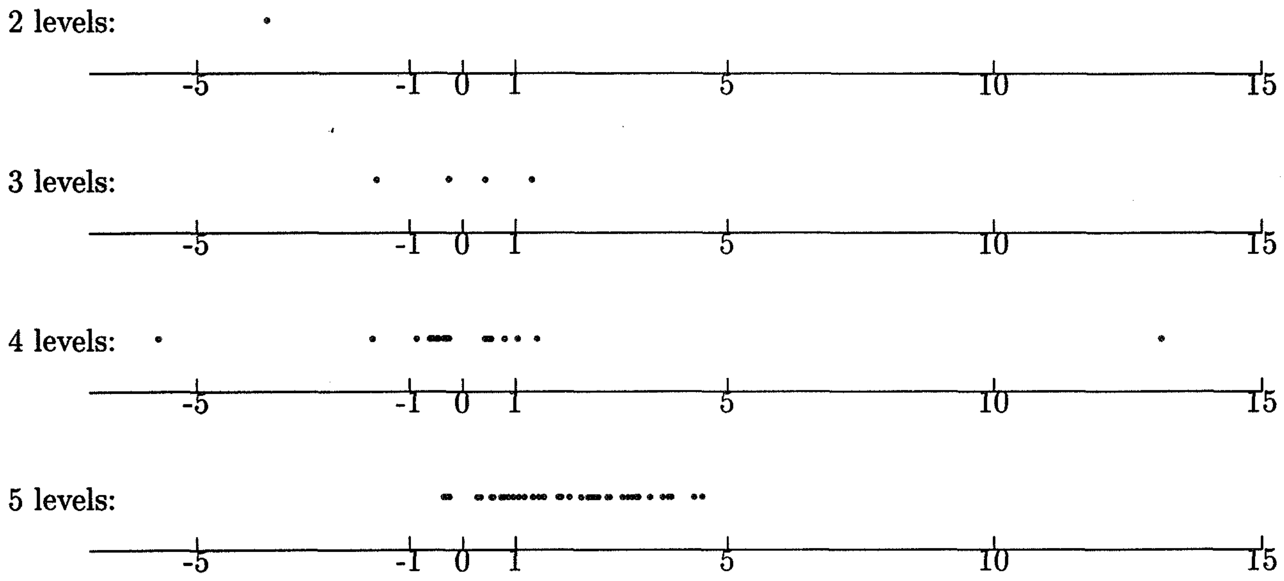


Figure 1: Eigenvalues (of magnitude ≥ 0.25) of the PAMUG iteration matrix for the indefinite Helmholtz equation with $\beta = 3200$, $N = 256$ and periodic boundary conditions.

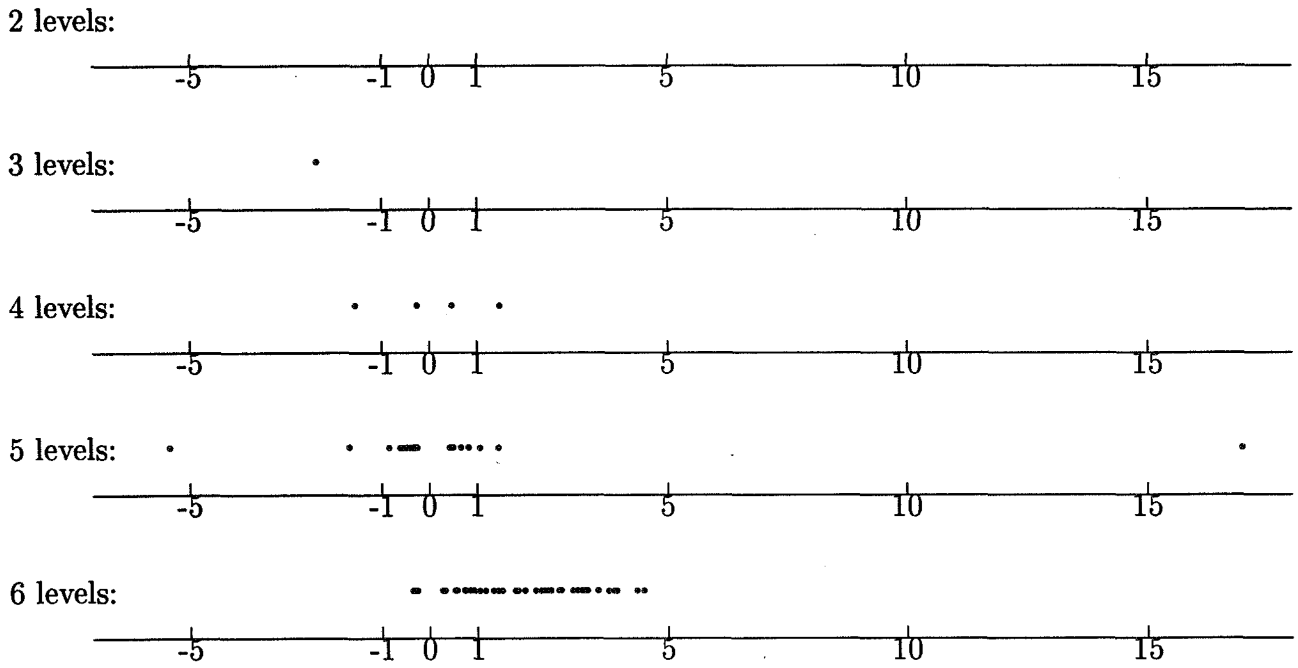


Figure 2: Eigenvalues (of magnitude ≥ 0.25) of the PAMUG iteration matrix for the indefinite Helmholtz equation with $\beta = 3200$, $N = 512$ and periodic boundary conditions.

4 NUMERICAL EXPERIMENTS

4.1 A Comparison of Various Multigrid Methods

We apply AutoMUG and several other multigrid algorithms to the problem

$$-u_{xx} - u_{yy} - 800u = f, \quad (x,y) \in \Omega = (0,1) \times (0,1),$$

with complex boundary conditions of the third kind

$$\tfrac{\partial u}{\partial n} + 10iu = g \quad (x,y) \in \Gamma \subset \partial\Omega$$

(where $\vec{n}$ is the outer normal vector) and Dirichlet boundary conditions on $\partial\Omega \setminus \Gamma$. We consider the following cases:

$$\begin{array}{ll} \text{(a)} & \Gamma = \emptyset \\ \text{(b)} & \Gamma = \{0\} \times [0,1]. \end{array}$$

The equation is discretized via a second-order five-point difference scheme (as in (3)–(4)). Uniform $N \times N$ grids are used. The exact solution is $u = xy$. The initial guess is random in (0,1).

To the basic multi-level iteration (2), we apply the Transpose Free Quasi Minimal Residual (TFQMR) acceleration method (Algorithm 5.2 in [12]), which avoids the computation of the transpose of the coefficient matrix and preconditioner (the latter is only implicitly given in (1), so its transpose is not available). TFQMR may be considered a modification of the Conjugate Gradient Squared (CGS) method of [19]. The costs of these acceleration techniques are comparable to that of the Conjugate Gradient method, that is, about 1–1.5 work units per iteration. We found that the performance of CGS and TFQMR is similar; we preferred the latter, though, because of its smooth convergence curve.

The multi level methods are implemented with the red-black Gauss-Seidel (RB) smoother in a V(1,1)-cycle. The coarsest level equation is solved with six orders of magnitude accuracy.

We define the following measures of efficiency: the convergence factor

$$\text{cf} = \frac{||Ax_{last} - b||_2}{||Ax_{last-1} - b||_2}$$

and the averaged convergence factor

$$\text{avcf} = \left(\frac{||Ax_{last} - b||_2}{||Ax_0 - b||_2} \right)^{1/last},$$

where *last* is the smallest positive integer for which

$$\frac{||Ax_{last} - b||_2}{||Ax_0 - b||_2} \leq \text{threshold}$$

and threshold is about 10^{-6}. When acceleration is used, the convergence factor often oscillates; hence, for the highly indefinite examples, only avcf is reported.

AutoMUG is compared to 3 other multigrid methods which share the same complexity (that is, use 5-coefficient stencils at all levels):

1. Standard Multigrid (MG): coarse grid operators are derived from rediscretizations of the differential equation; full-weighting and bilinear interpolation are used for restriction and prolongation, respectively.
2. Cyclic Reduction Multigrid (CR-MG) [8]: coarse grid operators, restriction and prolongation are defined as in [8].
3. Full CR-MG (F-CR-MG): coarse grid operators are generated from [8]; full-weighting and bilinear interpolation are used for restriction and prolongation, respectively.

The results are displayed in Tables 1 and 2.

Table 1: Averaged convergence factors (avcf) for various multigrid methods (with TFQMR acceleration). The results show that once the resolution of the coarsest grid is fixed, the rate of convergence is independent of the number of levels.

N	levels	Γ	MG	F-CR-MG	CR-MG	AutoMUG
255	4	(a)	.540	.267	.614	.277
127	3	(a)	.549	.272	.506	.280
63	2	(a)	.561	.273	.404	.312
63	2	(b)	.651	.694	.748	.396

Table 2: Averaged convergence factors (with TFQMR acceleration) showing the deterioration of convergence rates when the resolution of the coarsest grid is too coarse.

N	levels	Γ	MG	F-CR-MG	CR-MG	AutoMUG
63	2	(a)	.561	.273	.404	.312
63	3	(a)	$> .9$	.771	$> .95$	.737

Remark: it was also found that for diffusion problems with discontinuous coefficients (e.g., Examples 7 and 9 in [18]) MG and both variants of MG-CR stagnate.

4.2 Problems with Discontinuous Coefficients

AutoMUG and two variants of Black Box Multigrid are applied to problems of the form

$$-\nabla(D\nabla u) - \sigma u = f \quad \text{in } \Omega \equiv (0,\omega_2) \times (0,\omega_2),$$

with

$$\begin{aligned} j(t) &= \begin{cases} 0 & 0 < t < \omega_1 \\ 1 & \omega_1 < t < \omega_2 \end{cases}, \\ D(x,y) &= \begin{cases} d_r & (x,y) \in \Omega, \quad j(x) + j(y) \bmod 2 = 0 \\ d_b & (x,y) \in \Omega, \quad j(x) + j(y) \bmod 2 = 1 \\ d_o & \qquad\qquad\quad (x,y) \notin \Omega \end{cases}, \end{aligned}$$

$$\sigma(x,y) = \begin{cases} \sigma_r & (x,y) \in \Omega, \quad j(x)+j(y) \bmod 2 = 0 \\ \sigma_b & (x,y) \in \Omega, \quad j(x)+j(y) \bmod 2 = 1 \\ \sigma_o & (x,y) \notin \Omega \end{cases},$$

$$f(x,y) = \begin{cases} 0 & (x,y) \in \Omega, \quad j(x)+j(y) \bmod 2 = 0 \\ 1 & (x,y) \in \Omega, \quad j(x)+j(y) \bmod 2 = 1 \\ 0 & (x,y) \notin \Omega \end{cases}$$

and mixed boundary conditions of the form

$$\begin{array}{ll} Du_n + \gamma_0 u = 0 & x = 0 \text{ or } y = 0 \\ Du_n + \gamma_1 u = 0 & x = \omega_2 \text{ or } y = \omega_2 \end{array}$$

(where ω_1, ω_2, γ_0, γ_1, d_r, d_b, d_o, σ_r, σ_b and σ_o are parameters). The finite volume discretization of [2] is used. However, since it results in a strong coupling between domains which are only weakly coupled in the PDE and, hence, in an inadequate scheme (see [2]), it is not applied to the original but to the modified problem $-\nabla(\tilde{D}\nabla u) - \tilde{\sigma}u = f$, where

$$\begin{aligned} \varepsilon &= \frac{d_r + d_b}{2} \min(d_r/d_b, d_b/d_r) \\ \delta &= \frac{\sigma_r + \sigma_b}{2} \min(d_r/d_b, d_b/d_r) \\ \tilde{D}(x,y) &= \begin{cases} \varepsilon & |x-\omega_1| + |y-\omega_1| \le h \\ D(x,y) & \text{otherwise} \end{cases} \\ \tilde{\sigma}(x,y) &= \begin{cases} \delta & \max(|x-\omega_1|, |y-\omega_1|) \le h/2 \\ \sigma(x,y) & \text{otherwise.} \end{cases} \end{aligned}$$

A uniform 63 × 63 fine grid is used (the only exception to this are Examples (12)–(13) in Table 3 representing the 'staircase' problem of [2], where a uniform 17 × 17 fine grid is used). When Dirichlet boundary conditions are imposed, it is denoted by $\gamma_0 = \gamma_1 = \infty$. In this case, no grid point lies on $\partial\Omega$; all equations are non-trivial. The initial guess is zero.

The results in Table 3 correspond to the following methods: (A) AutoMUG; (B) Black Box Multigrid [9]; and (C) the second method in [10]. For Examples (1)-(11), these methods were implemented with coarse grids consisting of even numbered variables of the next finer grid (similar results, however, were obtained when odd numbered variables were used for this purpose). The off diagonal row-sum modification introduced in [10] is not used, since (apart from Examples (12)–(13)) coarse grids do not include boundary points of the next finer grid (see [15]). Also, prolongation is done without using the right hand side, since it was found in [15] that this does not improve the convergence for indefinite problems.

The multigrid cycle is implemented as in the previous subsection. For methods (B) and (C), however, since 9-coefficient stencils are used, RB is replaced by the four-color ordering of [1]. Acceleration is used only for highly indefinite problems, namely, when $\max(\sigma_r, \sigma_b) > 100$.

A comparison of Examples (1) and (2) of Table 3 shows that, as implied by Corollary 1, AutoMUG (with no acceleration) performs for nearly singular Helmholtz equations almost as well as for the Poisson equation. For more highly indefinite problems, however, acceleration must be used.

Table 3: Three multigrid methods, (A) AutoMUG, (B) Black Box Multigrid and (C) the second method of Dendy (87), applied to definite and indefinite problems with discontinuous coefficients. Uniform 63 × 63 (resp., 17 × 17) fine grids are used for Examples (1)-(11) (resp., (12)-(13), the 'staircase' problem).

Description of examples

example	ω_1	ω_2	γ_0	γ_1	d_r	d_b	d_o	σ_r	σ_b	σ_o	acceleration
(1)		1	∞	∞	1	1	1	0	0	0	no
(2)		1	∞	∞	1	1	1	20	20	20	no
(3)		1	∞	∞	1	1	1	400	400	400	yes
(4)		1	$10i$	$10i$	1	1	0	400	400	0	yes
(5)	30/62	1	$10i$	$10i$	1	1	0	0	400	0	yes
(6)	31/62	1	$10i$	$10i$	1	1	0	0	400	0	yes
(7)	30/62	1	$10i$	$10i$	1000	1	0	0	400	0	yes
(8)	31/62	1	$10i$	$10i$	1000	1	0	0	400	0	yes
(9)		62	0	0.5	1	1	0	0	0	0	no
(10)	30	62	0	0.5	1000	1	0	0	0	0	no
(11)	31	62	0	0.5	1000	1	0	0	0	0	no

Numerical results

			cf			avcf	
example	levels	A	B	C	A	B	C
(1)	4	.095	.065	.159	.090	.072	.184
(2)	4	.096	.431	> 1	.091	.507	> 1
(3)	3				.336	.702	.835
(4)	3				.329	.335	.567
(5)	3				.369	.315	.516
(6)	3				.295	.285	.464
(7)	3				.298	.283	> .8
(8)	3				.291	.341	.530
(9)	4	.160	.118	.238	.151	.114	.267
(10)	4	.381	.120	.211	.429	.142	.232
(11)	4	.148	.987	.988	.192		
(12)	2	.153	.121	.133	.196	.141	.151
(13)	3	> 1	.220	.240	> 1	.237	.269

Examples (9)–(13) deal with diffusion problems with discontinuous coefficients. In particular, Examples (12)-(13) are the 'staircase' problem (Example IV in [2], where $D = 1000$ inside the staircase and $D = 1$ outside).

It is evident from Example (11) that Black Box Multigrid stagnates when the break point ω_1 lies on the coarse grids. The reason for this is that the 9-coefficient stencils of its coarse grid operators involve strong coupling between domains which are only weakly coupled in the PDE. Hence, in this case, the 5-coefficient stencils of AutoMUG are preferable (see [15] for a variant of Black Box Multigrid which overcomes this problem).

It is interesting to mention that when D, rather than $\tilde{D}$, is used for the finite volume discretization in Example (10), Black Box Multigrid converges rapidly while AutoMUG diverges. However, in light of the remarks made in [2], it is not clear whether the resulting scheme is meaningful.

Acknowledgment. The author wishes to thank Moshe Israeli for suggesting the physical motivation for the restriction on the grid resolution and Irad Yavneh for his valuable comments.

APPENDIX A

Proof of Theorem 1: Let $\vec{v}$ be a common eigenvector of X, Y and A with the eigenvalues x, y and $x+y$, respectively. Then $\vec{v}$ is also an eigenvector of the iteration matrix of PAMUG(0) with the corresponding eigenvalue $f_r(x,y)$, where

$$g(x,y) = 1 - \frac{(2-x)(2-y)(x+y)}{2(x(2-x)+y(2-y))} = \frac{xy(4-x-y)}{2(x(2-x)+y(2-y))}$$

$$\text{and } f_r(x,y) = \left(1 - \frac{x+y}{3}\right)^r g(x,y).$$

To prove the theorem, it is sufficient to bound $|f_r|$ in the region $0 < x, y < 2$. In this region, $0 < |f_r| < g < 1$. Since g is symmetric, it is natural to write it as a function of the symmetric variables $c = x + y$ and $d = xy$. Clearly, $(c,d) \in (0,4) \times (0,4)$,

$$g(c,d) = \frac{d(4-c)}{2(2c-c^2+2d)} \quad \text{and} \quad f_r(c,d) = \left(1 - \frac{c}{3}\right)^r g(c,d).$$

The partial derivative of g with respect to d is

$$\begin{aligned} \frac{\partial g}{\partial d}(c,d) &= \frac{(4-c)(2c-c^2+2d) - 2d(4-c)}{2(2c-c^2+2d)^2} \\ &= \frac{(4-c)c(2-c)}{2(2c-c^2+2d)^2}. \end{aligned}$$

Hence $\partial g/\partial d > 0$ if $0 < c < 2$, $\partial g/\partial d = 0$ if $c = 2$ and $\partial g/\partial d < 0$ if $2 < c < 4$. Assume that $0 < c < 2$. Then g achieves its maximum on the hyperbola $xy = d$ for which d is maximal. This happens at the point $x = y = c/2$. But at this point we have $g = c/4$ and

$$f_r = \left(\frac{3-c}{3}\right)^r \frac{c}{4} \equiv h(c)$$

We find the maxima of h:

$$h'(c) = \left(\frac{1}{c} - \frac{r}{3-c}\right) h(c) = 0$$

or $3 - c - cr = 0$ or $c = 3/(r+1)$. The maximum of h in $(0, 2)$ is thus

$$h\left(\frac{3}{r+1}\right) = \frac{3}{4} \frac{r^r}{(r+1)^{r+1}}.$$

The theorem follows from $|f_r| \leq \left(\frac{3-2}{3}\right)^r = 3^{-r}$ in the region $2 \leq c < 4$. □

APPENDIX B

Proof of Corollary 1: For $i \in \{0, 1\}$, define the injections $O_{x,i}$ and $O_{y,i}$ by

$$(O_{x,i}v)_{l,m} = \begin{cases} v_{l,m} & l = i \bmod 2 \\ 0 & l \neq i \bmod 2 \end{cases} \quad \text{and} \quad (O_{y,i}v)_{l,m} = \begin{cases} v_{l,m} & m = i \bmod 2 \\ 0 & m \neq i \bmod 2 \end{cases}, \quad v \in V_N$$

($O_{x,i}$ injects onto every other y-line and $O_{y,i}$ injects onto every other x-line). Let v be a common eigenvector of X and Y with the corresponding eigenvalues x_v and y_v, respectively. Since X and Y are of property–A, it follows from [21], Sec. 7.1 that the following is a set of common eigenvectors of X and Y:

$$W \equiv \left\{ \sum_{i,j \in \{0,1\}} (-1)^{\alpha i + \beta j} O_{x,i} O_{y,j} v \right\}_{\alpha, \beta \in \{0,1\}}$$

The elements of W are orthogonal to each other and have the same l_2 norm. Denote by x_w (resp., y_w) the eigenvalue of an element $w \in W$ with respect to X (resp., Y). Define the set of vectors

$$V \equiv \{2 O_{x,i} O_{y,j} v\}_{i,j \in \{0,1\}}.$$

Define the symmetric orthogonal discrete Haar transform

$$H = (h_{\gamma,\delta})_{\gamma,\delta \in \{0,1\}^2}, \qquad h_{\gamma,\delta} = 2^{-1}(-1)^{\sum_{i=1}^{2} \gamma_i \delta_i}.$$

Hence $W = HV$ and $V = HW$. Let M_A and M_P denote the iteration matrices of AutoMUG(0) and PAMUG(0), respectively. Note that $O^T O = O_{x,0} O_{y,0}$ and that $OM_A = OM_P$. The assumption that a postsmoothing of the form $x \leftarrow POx$ is performed is equivalent to replacing the substitution $x_{out} \leftarrow x_{in} - Pe$ in (1) by $x_{out} \leftarrow P(Ox_{in} - e)$. From these observations and the proof of Theorem 1, it follows that, for any $w \in W$,

$$M_A w = f_r(x_w, y_w) \sum_{i,j \in \{0,1\}} (1 - x_v)^i (1 - y_v)^j O_{x,i} O_{y,j} v.$$

Consequently, $span(W)$ is an invariant subspace of M_A. Let $\hat{M}_A$ denote the restriction of M_A to $span(W)$. The representation of $\hat{M}_A$ in the basis W is of the form $\hat{M}_A = 2^{-1} H p u^t$, where p and u are the following four-dimensional vectors:

$$p = (1, 1 - x_v, 1 - y_v, (1 - x_v)(1 - y_v))^t \quad \text{and} \quad u = (f_r(x_w, y_w))^t_{w \in W}.$$

Let ρ denote the spectral radius of a matrix. Then

$$\|\hat{M}_A\| \leq \frac{1}{2}\rho(pu^tup^t)^{1/2} = \frac{1}{2}\|p\| \; \|u\| \leq \|u\| \leq 2\max_{w\in W}|f_r(x_w, y_w)|. \qquad \square$$

APPENDIX C

Proof of Theorem 2: Let

$$A_0 = A \quad \text{and} \quad D_i \equiv diag(A_i), \; 0 \leq i \leq n-1.$$

Consider the ith call to the PAMUG procedure in the PAMUG method (1), $1 \leq i \leq n$. This call is designated to solve the equation $A_{i-1}\vec{e} = \vec{r}$. For this equation, denote the two-level PAMUG iteration matrix by N_{i-1} and the multi-level PAMUG iteration matrix by M_{i-1}. For a PAMUG cycle with index ϵ, we have (see [13]) $M_{n-1} = N_{n-1}$, and, for $0 \leq i < n-1$,

$$\begin{aligned} M_i &= \left(I - (I - M_{i+1}^{\epsilon})A_{i+1}^{-1}R_{i+1}A_i\right)\left(I - \alpha^{-1}D_i^{-1}A_i\right)^r \\ &= N_i + M_{i+1}^{\epsilon}A_{i+1}^{-1}R_{i+1}A_i\left(I - \alpha^{-1}D_i^{-1}A_i\right)^r. \end{aligned}$$

It is easily seen by induction that all the operators X_i, $R(X_i)$, UY_iU and $UR(Y_i)U$, for every i, are block diagonal with circulant Toeplitz blocks. Hence, all the operators A_i, D_i and R_i, for every i, are diagonalizable by the 2-dimensional discrete Fourier transform; hence, so are also the operators N_i and, by induction, also the operators M_i. The theorem follows from spectral analysis. □

References

[1] Adams, L. M.; and Jordan, H. F.: Is SOR Color-Blind? *SIAM J. Sci. Stat. Comput.,* 7 (1986), pp. 490-506.

[2] Alcouffe, R.; Brandt, A.; Dendy, J. E.; and Painter J.: The Multigrid Method for the Diffusion Equation with Strongly Discontinuous Coefficients. *SIAM J. Sci. Stat. Comput.,* vol. 2, 1981, pp. 430-454.

[3] Bramble, J. H.; Leyk, Z.; and Pasciak, J. E.: Iterative Schemes for Non-Symmetric and Indefinite Elliptic Boundary Value Problems. *Math. Comp.* 60 (1993), pp. 1-22.

[4] Brandt, A.: Guide to Multigrid Development. In Multigrid Methods, Hackbusch, W.; and Trottenberg, U. *(eds.)*: *Lecture Notes in Mathematics* 960, Springer-Verlag, Berlin, Heidelberg 1982.

[5] Brandt, A.; and Ta'asan, S.: Multigrid methods for nearly singular and slightly indefinite problems. In Multigrid Methods ii, Proceedings, Cologne, 1985, *Lecture Notes in Mathematics* 1228, Springer-Verlag, Hackbusch, W.; and Trottenberg, U. *(eds.)*: pp. 100-122.

[6] Brandt, A.; and Yavneh, I.: On Multigrid Solution of High Reynolds Incompressible Entering Flows, *J. Comp. Phys.* 101, 1992, pp. 151-164.

[7] Brandt, A.; and Yavneh, I.: Accelerated Multigrid Convergence and High Reynolds Recirculating Flows, *SIAM J. Sci. Stat. Comput.* 14, 1993, pp. 607-626.

[8] Brackenridge, K.: Multigrid and Cyclic Reduction Applied to the Helmholtz Equation, Sixth Copper Mountain Conference on Multigrid Methods, Melson, N. D., McCormick, S. F. and Manteuffel, T. A. (eds.), NASA, Langley Research Center, Hampton, VA (1993), pp. 31-42.

[9] Dendy, J. E.: Black Box Multigrid. *J. Comp. Phys.*, vol. 48, 1982, pp. 366-386.

[10] Dendy, J. E.: Two Multigrid Methods for the Three-Dimensional Problems with Discontinuous and Anisotropic Coefficients. *SIAM J. Sci. Stat. Comput.*, 8 (1987), pp. 673-685.

[11] Frederickson P. O.; and McBryan O. A.: Parallel Superconvergent Multigrid. In Multigrid Methods, *Lecture Notes in Pure and Applied Mathematics* 110, McCormick, S. F. *ed.*, Marcel Dekker, N.Y., 1988.

[12] Freund R. W.: Transpose Free Quasi-Minimal Residual Algorithm for Non-Hermitian Linear Systems. *SIAM J. Sci. Stat. Comput.* 14 (1993), pp. 470-482.

[13] Hackbusch, W.: Multigrid Methods and Applications. Springer-Verlag, Berlin, Heidelberg, 1985.

[14] Saad, Y.; and Schultz, M. H.: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* 7, 1986, pp. 856-869.

[15] Shapira, Y.: Two-Level Analysis and Multigrid Methods for SPD, Non-Normal and Indefinite Problems. Technical Report #824 (revised version), Computer Science Department, Technion – Israel Institute of Technology, July 1994. submitted to *SIAM J. Sci. Comput.*.

[16] Shapira, Y.: Multigrid Methods for 3-d Definite and Indefinite Problems. Technical Report #834 (revised version), Computer Science Department, Technion – Israel Institute of Technology, Oct. 1994.

[17] Shapira, Y.; Israeli, M.; and Sidi, A.: An automatic Multigrid method for the solution of sparse linear systems. Sixth Copper Mountain Conference on Multigrid Methods, Melson, N. D., McCormick, S. F. and Manteuffel, T. A. (eds.), NASA, Langley Research Center, Hampton, VA (1993), pp. 567-582.

[18] Shapira, Y.; Israeli, M.; and Sidi, A.: Towards Automatic Multigrid Algorithms for SPD, Nonsymmetric and Indefinite Problems. *SIAM J. Sci. Comput.* to appear in March 1996.

[19] Sonneveld, P.; Wesseling, P.; and de Zeeuw, P. M.: Multigrid and Conjugate Gradient Methods as Convergence Acceleration Techniques. In Multigrid Methods for Integral and Differential Equations, Paddon, D.J., Holstein, H. (eds.), Oxford Univ. Press (1985), 117-168.

[20] Varga, R.: *Matrix Iterative Analysis.* Prentice-Hall, N. J., 1962.

[21] Young, D.: Iterative Solution of Large Linear Systems. Academic Press, N.Y., 1971.

Page intentionally left blank

A GENUINELY TWO-DIMENSIONAL SCHEME FOR THE COMPRESSIBLE EULER EQUATIONS*

David Sidilkover
ICASE, Mail Stop 132C
NASA Langley Research Center
Hampton, VA 23681

SUMMARY

We present a new genuinely multidimensional discretization for the compressible Euler equations. It is the only high-resolution scheme known to us where Gauss-Seidel relaxation is stable when applied as a smoother directly to the resulting high-resolution scheme. This allows us to construct a very simple and highly efficient multigrid steady-state solver. The scheme is formulated on triangular (possibly unstructured) meshes.

INTRODUCTION

One of the most challenging problems in numerical analysis was the construction of a numerical scheme for gas dynamics in one dimension. Such a scheme had to combine high-order accuracy in the regions of the smooth flow with the ability to represent discontinuities by thin oscillation-free layers. These two properties are not both attainable within the class of linear schemes (Godunov's theorem). Therefore, the successful scheme should be non-linear. Schemes of this type were named high-resolution schemes. The discrete schemes for the equations of gas dynamics in multidimensions are usually obtained using the dimensional-splitting approach, i.e. applying a one-dimensional scheme in each coordinate direction. The main problem, however, is that the steady-state solvers based on such schemes suffer from poor computational efficiency. It was observed by Spekreijse [1] that such a simple and efficient smoother as pointwise Gauss Seidel relaxation is unstable in conjunction with such schemes even in the simple case of linear advection equation. The multigrid solvers, therefore, have to resort to multi-stage Runge-Kutta relaxation or to defect-correction techniques, which are not the really efficient ways to utilize the multigrid approach.

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480.

The reason for the fact that the Gauss-Seidel relaxation is unstable when applied in conjunction with the dimensionally-split high resolution schemes can be traced down to the particular way the nonlinearity is incorporated within these schemes. This motivated the search for a high resolution (at least at the steady-state) scheme, with the nonlinear high-resolution correction introduced in such a way that it does not lead to the instability of the Gauss-Seidel relaxation. This search resulted in the genuinely multidimensional advection scheme of the control volume type (see [2],[3]). The so-called fluctuation-splitting type schemes (for unstructured triangular meshes) were also introduced (see [4],[5]). A strong relationship between the two types was established in [6]. However, it was not clear for a long time how to extend these ideas to the systems of equations. One of the major directions was the so-called wave modeling (see [7],[8]). This approach concentrated on finding a way to represent (locally) the physics of two-dimensional flow of a compressible fluid by a finite number of simple waves, each one having an associated advection equation. However, numerical schemes created this way suffered from a lack of robustness. The approach introduced in [9] is concerned not with applying an advection scheme to discretize a system of equations in two dimensions, but rather with applying to the systems of equations the same strategy that was used when constructing a scalar advection scheme. The resulting genuinely two-dimensional scheme is formulated on triangular (possibly unstructured) meshes. The unique advantage of this high-resolution discretization is that the Collective Gauss-Seidel relaxation can be applied directly to the high resolution discrete equations. This results in a very simple and efficient multigrid steady-state solver.

In this paper first we introduce some further enhancements to the scheme presented in [9]. Numerical experiments will be presented. Some possible extensions of the truly multidimensional approach will be discussed.

GENUINELY TWO-DIMENSIONAL ADVECTION SCHEME

Consider a linear two-dimensional advection equation

$$u_t + au_x + bu_y = 0. \tag{1}$$

Consider the triangulation of the domain as illustrated on Fig.1. Denote by R the *fluctuation* (i.e., the residual of equation (1) on triangle T multiplied by the area of this triangle):

$$R = R^x + R^y, \tag{2}$$

where

$$\begin{aligned} R^x &= -\tfrac{h}{2}[a(u_0 - u_3)] \\ R^y &= -\tfrac{h}{2}[b(u_3 - u_4)]. \end{aligned}$$

The following fluctuation distribution formulae

$$\begin{aligned} h^2u_0^{n+1} &= h^2u_0^n + \tfrac{\tau}{2}R^x \\ h^2u_3^{n+1} &= h^2u_3^n + \tfrac{\tau}{2}[R^x + R^y] \\ h^2u_4^{n+1} &= h^2u_4^n + \tfrac{\tau}{2}R^y \end{aligned} \tag{3}$$

reproduce the central difference scheme, which is second-order accurate (in space) but is known to be unstable.

We shall introduce here the *positivity* property.

Definition 1. *A scheme is said to be of the* <u>*positive*</u> *type if any solution value on the new time level obtained by this scheme can be written as a positive combination of the values from the previous time level.*

Solutions obtained by using positive schemes satisfy a certain maximum principle and, therefore, do not exhibit oscillatory behavior in the presence of discontinuities. It is obvious that the central scheme (3) is not of the positive type.

Modifying (3) by adding the appropriate artificial viscosity terms

$$\begin{aligned} h^2 u_0^{n+1} &= h^2 u_0^n + \tfrac{\tau}{2}[R^x(1+\text{sign}(a))] \\ h^2 u_3^{n+1} &= h^2 u_3^n + \tfrac{\tau}{2}[R^x(1-\text{sign}(a)) + R^y(1+\text{sign}(b))] \\ h^2 u_4^{n+1} &= h^2 u_4^n + \tfrac{\tau}{2}[R^y(1-\text{sign}(b))] \end{aligned} \tag{4}$$

we recover the dimensional upwind scheme which is *positive*, but only first order accurate.

Definition 2. *The fluctuation-splitting scheme is called* <u>*linearity preserving*</u> *if whenever the fluctuation on the triangle T vanishes then the scheme leads to a zero update in each of the three vertices of the triangle.*

The upwind scheme (4) does not satisfy this property since the fact that $R = 0$ does not necessarily imply that $R^x = R^y = 0$. Therefore, a non-zero update of the nodal values may be introduced.

Introduce the following quantities

$$\begin{aligned} R^{x^*} &= R^x + R^y \Psi(Q) \\ R^{y^*} &= R^y + R^x \tfrac{\Psi(Q)}{Q} \end{aligned} \tag{5}$$

where

$$Q = -\frac{R^x}{R^y} \tag{6}$$

and Ψ is a Lipschitz continuous limiter function such that

$$0 \leq \Psi(Q) \leq 1, \quad 0 \leq \frac{\Psi(Q)}{Q} \leq 1 \tag{7}$$

and

$$\Psi(1) = 1. \tag{8}$$

Substituting R^{x^*}, R^{y^*} for R^x, R^y into (4) satisfies the linearity preserving property. This can be demonstrated in the following way: assume that $R = 0$. This means that $R^x = -R^y$ or $Q = -R^x/R^y = 1$. It can be seen that no update will be introduced to any of the unknowns at the nodes of triangle T, provided the limiter-function satisfies the equality (8). This scheme is also second order accurate at the steady-state, since the grid considered here is structured (see [6]).

Using the following identity

$$R^y\Psi(Q) \equiv -R^x\frac{\Psi(Q)}{Q} \tag{9}$$

we can rewrite (5) in the following form

$$\begin{aligned} R^{x^*} &= R^x(1 - \tfrac{\Psi(Q)}{Q}) \\ R^{y^*} &= R^y(1 - \Psi(Q)). \end{aligned} \tag{10}$$

It is easy to see that the scheme defined by (4) and (5) (or 10) is of positive type, provided the inequality (7) holds.

It is also obvious from (9) that such scheme is conservative because

$$R^{x^*} + R^{y^*} \equiv R^x + R^y \equiv R$$

(for more details see [6]).

MULTIDIMENSIONAL EULER SCHEME

The Euler equations of gas dynamics in two dimensions can be written

$$\boldsymbol{u}_t + \boldsymbol{F}(\boldsymbol{u})_x + \boldsymbol{G}(\boldsymbol{u})_y = \mathbf{o}, \tag{11}$$

where

$$\boldsymbol{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}; \quad \boldsymbol{F}(\boldsymbol{u}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}; \quad \boldsymbol{G}(\boldsymbol{u}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix} \tag{12}$$

where the enthalpy H is defined by

$$H = \frac{e+p}{\rho} = \frac{c^2}{\gamma - 1} + \frac{u^2+v^2}{2}, \tag{13}$$

the speed of sound

$$c = \sqrt{\frac{\gamma p}{\rho}} \tag{14}$$

and the pressure

$$p = (\gamma - 1)(e - \rho\frac{u^2+v^2}{2}). \tag{15}$$

The quasilinear non-conservative formulation of the Euler system in *auxiliary* variables (s, u, v, p) can be introduced in two dimensions as well

$$\begin{aligned} &s_t + us_x + vs_y = 0 \\ &\rho u_t + \rho uu_x + \rho vu_y + p_x = 0 \\ &\rho v_t + \rho uv_x + \rho vv_y + p_y = 0 \\ &p_t + up_x + vp_y + \rho c^2(u_x + v_y) = 0 \end{aligned} \tag{16}$$

where $ds = d\rho - \frac{dp}{c^2}$.

Remark 3. *Note that the entropy (s) evolution is subject to the two-dimensional advection equation, which is locally decoupled from the rest of the system.*

The fluctuation of the system (11) defined over the triangle T is

$$\boldsymbol{R} = \int\int \boldsymbol{u}_t = -\int\int (\boldsymbol{F}_x + \boldsymbol{G}_y)\, dx\, dy = -S_T\left[\widehat{\boldsymbol{F}}_x + \widehat{\boldsymbol{G}}_y\right] \tag{17}$$

where $\widehat{\boldsymbol{F}}_x, \widehat{\boldsymbol{G}}_y$ are some averaged values of the flux derivatives over the triangle T.

Our construction of the truly two-dimensional Euler scheme utilizes the two-dimensional conservative linearization procedure [10]. We assume that the quantity which varies linearly over an element is the "parameter vector"

$$\boldsymbol{m} = \sqrt{\rho}(1, u, v, H)^T \tag{18}$$

and its averaged value on the triangle T (as illustrated on Fig.1) is given by the following

$$\tilde{\boldsymbol{m}} = \frac{\boldsymbol{m}_0 + \boldsymbol{m}_3 + \boldsymbol{m}_4}{3} \tag{19}$$

Roe-averaged quantities can be introduced

$$\begin{aligned} \tilde{u} &= \tilde{m}_2/\tilde{m}_1 \\ \tilde{v} &= \tilde{m}_3/\tilde{m}_1 \\ \tilde{H} &= \tilde{m}_4/\tilde{m}_1 \end{aligned} \tag{20}$$

and

$$\tilde{c}^2 = (\gamma - 1)[\tilde{H} - \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2)]. \tag{21}$$

Fluctuations of the Euler system in the *auxiliary* variables can be presented as

$$\boldsymbol{r} = \boldsymbol{r}^x + \boldsymbol{r}^y, \tag{22}$$

where

$$\begin{aligned} \boldsymbol{r}^x &= -S_T\tilde{A} \cdot (\widehat{s_x}, \widehat{\rho u_x}, \widehat{\rho v_x}, \widehat{p_x})^T \\ \boldsymbol{r}^y &= -S_T\tilde{B} \cdot (\widehat{s_y}, \widehat{\rho u_y}, \widehat{\rho v_y}, \widehat{p_y})^T \end{aligned}$$

with

$$\tilde{A} = \begin{pmatrix} \tilde{u} & 0 & 0 & 0 \\ 0 & \tilde{u} & 0 & 1 \\ 0 & 0 & \tilde{u} & 0 \\ 0 & \tilde{c}^2 & 0 & \tilde{u} \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \tilde{v} & 0 & 0 & 0 \\ 0 & \tilde{v} & 0 & 0 \\ 0 & 0 & \tilde{v} & 1 \\ 0 & 0 & \tilde{c}^2 & \tilde{v} \end{pmatrix}$$

and $S_T = h^2/2$ is the area of the triangle T, and

$$\widehat{\rho_x} = 2\tilde{m}_1(m_1)_x \tag{23}$$

$$\widehat{\rho u_x} = \tilde{m}_1(m_2)_x - \tilde{m}_2(m_1)_x \tag{24}$$

$$\widehat{\rho v_x} = \tilde{m}_1(m_3)_x - \tilde{m}_3(m_1)_x \tag{25}$$

$$\widehat{p_x} = \frac{\gamma - 1}{\gamma}[(\tilde{m}_4(m_1)_x + \tilde{m}_1(m_4)_x) + (\tilde{m}_2(m_2)_x + \tilde{m}_3(m_3)_x)]. \tag{26}$$

The corresponding terms involving derivatives in the y direction can be written in the analogous manner.

Introducing the matrix

$$C_a = \begin{pmatrix} 1 & 0 & 0 & 1/\tilde{c}^2 \\ \tilde{u} & 1 & 0 & \tilde{u}/\tilde{c}^2 \\ \tilde{v} & 0 & 1 & \tilde{v}/\tilde{c}^2 \\ (\tilde{u}^2+\tilde{v}^2)/2 & \tilde{u} & \tilde{v} & 1/(\gamma-1)+(\tilde{u}^2+\tilde{v}^2)/(2\tilde{c}^2) \end{pmatrix} \tag{27}$$

we can define

$$\begin{aligned} \boldsymbol{R}^x &= C_a \boldsymbol{r}^x \\ \boldsymbol{R}^y &= C_a \boldsymbol{r}^y. \end{aligned} \tag{28}$$

It can be easily verified that

$$\begin{aligned} \boldsymbol{R}^x &= -S^T \widehat{\boldsymbol{F}_x} \\ \boldsymbol{R}^y &= -S^T \widehat{\boldsymbol{G}_y}, \end{aligned} \tag{29}$$

where $\widehat{\boldsymbol{F}_x}, \widehat{\boldsymbol{G}_y}$ are the same averaged flux derivative values as defined in [10]. It is also obvious that the entire fluctuation

$$\boldsymbol{R} = \boldsymbol{R}^x + \boldsymbol{R}^y = C_a(\boldsymbol{r}^x + \boldsymbol{r}^y) = C_a \boldsymbol{r}. \tag{30}$$

Consider triangle T as illustrated in Fig.1. The fluctuation is distributed according to the following formulae:

$$\begin{aligned} S\boldsymbol{u}_0^{n+1} &= S\boldsymbol{u}_0^n & &+\tfrac{\tau}{2}C_a[\boldsymbol{r}^x(I-\text{sign}(\tilde{A}))] \\ S\boldsymbol{u}_3^{n+1} &= S\boldsymbol{u}_3^n & &+\tfrac{\tau}{2}C_a[\boldsymbol{r}^x(I+\text{sign}(\tilde{A}))+\boldsymbol{r}^y(I-\text{sign}(\tilde{B}))] \\ S\boldsymbol{u}_4^{n+1} &= S\boldsymbol{u}_4^n & &+\tfrac{\tau}{2}C_a[\boldsymbol{r}^y(I+\text{sign}(\tilde{B}))] \end{aligned} \tag{31}$$

we obtain the scheme that is similar to the standard Roe dimensionally split scheme. The only difference is in the linearization procedure.

We can construct now a (linearity preserving) second order accurate scheme. First, we shall introduce vectors $\boldsymbol{r}^{x^*}, \boldsymbol{r}^{y^*}$ with their elements defined by

$$\begin{aligned} r_i^{x^*} &= r_i^x + \Psi(q_i) r_i^y \\ r_i^{y^*} &= r_i^y + \tfrac{\Psi(q_i)}{q_i} r_i^x \end{aligned} \tag{32}$$

for $i = 1,2,3,4$, where

$$q_i = -\frac{r_i^x}{r_i^y} \tag{33}$$

and Ψ is a (non-compressive) limiter.

Substituting $\boldsymbol{r}^{x^*}, \boldsymbol{r}^{y^*}$ for $\boldsymbol{r}^x, \boldsymbol{r}^y$ in (31) we obtain a genuinely two-dimensional scheme, which is also linearity preserving (second order accurate in this case) and conservative.

Some attributes and properties of the genuinely multidimensional schemes will be discussed later in [9]. In order to obtain an efficient implementation of the scheme described above, it is important to write down the explicit expressions for the matrices $\text{sign}(\tilde{A}), \text{sign}(\tilde{B})$. Denote

$$M_x = \text{sign}(\tilde{A})$$
$$M_y = \text{sign}(\tilde{B}).$$

For matrix M_x the distinction should be made between two cases

$$M_x = \begin{cases} M_x^{sub}, & \text{if } |\tilde{u}| \leq \tilde{c} \\ M_x^{sup}, & \text{if } |\tilde{u}| > \tilde{c}, \end{cases} \tag{34}$$

and similarly

$$M_y = \begin{cases} M_y^{sub}, & \text{if } |\tilde{v}| \leq \tilde{c} \\ M_y^{sup}, & \text{if } |\tilde{v}| > \tilde{c}, \end{cases} \tag{35}$$

where

$$M_x^{sup} = \text{sign}(\tilde{u})I, \tag{36}$$

$$M_y^{sup} = \text{sign}(\tilde{v})I \tag{37}$$

and I is the 4×4 unity matrix. These matrices for the subsonic case appear to be surprisingly simple as well

$$M_x^{sub} = \begin{pmatrix} \text{sign}(\tilde{u}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\tilde{c} \\ 0 & 0 & \text{sign}(\tilde{u}) & 0 \\ 0 & \tilde{c} & 0 & 0 \end{pmatrix}, \tag{38}$$

$$M_y^{sub} = \begin{pmatrix} \text{sign}(\tilde{v}) & 0 & 0 & 0 \\ 0 & \text{sign}(\tilde{v}) & 0 & 0 \\ 0 & 0 & 0 & 1/\tilde{c} \\ 0 & 0 & \tilde{c} & 0 \end{pmatrix}. \tag{39}$$

Their structure indicates that there are some intriguing similarities between the standard schemes used for incompressible flow computations and the multidimensional upwind scheme presented above (see [9]).

Remark 4. *The scheme formulated here can be extended to the case of general unstructured grids in a straightforward way. Having a general triangular element, one has to introduce a new (possibly non-orthogonal) coordinate system whose axes align with two chosen faces of this element (Fig.2). The Euler system has to be rewritten in these new coordinates. Then one can follow directly the procedure of constructing the fluctuation distribution formulae presented in this section (see [9] for more details).*

NUMERICAL EXPERIMENTS

The purpose of the numerical experiments reported in this section is to verify the robustness of the constructed scheme and the quality of the numerical solutions obtained by its means. Some experiments illustrating the performance of the multigrid algorithm using this scheme are presented as well.

Supersonic flow in a channel with a bump

The test case considered here is a supersonic (Mach=2.9) flow in a channel with a circular bump. The bump is located at the lower wall of the channel at $1 \leq x \leq 2$, and its surface is a circular arch of $\pi/3$ and radius 1. Note that the actual shape of the domain is a rectangle. The influence of the bump on the flow is imposed through the boundary conditions: the velocity component normal to the surface of the bump at a certain location is being reflected.

The first experiment uses a grid of size 200×40 points. The density contour plots of the steady-state solution are presented on Fig.3(a). The scheme used is the one given by (31), (32) with the *minmod* limiter.

The second experiment presented in Fig.3(b) corresponds to the same settings, except that the grid is twice finer (400×80 points). As is expected, the grid refinement results in a better resolution of the flow features.

Transonic flow over a circular bump

The test case considered here is a transonic flow (free-stream Mach= .9) over a flat wall with a bump (Fig.4). The surface of the bump is a circular arch of $\pi/3$ and radius 1 and its location is between $3.5 \leq x \leq 4.5$. Again, in order to keep the experiments simple at this stage of work, the bump is treated the same way as in the previous experiments. The grid is 200×200 points. The shock of the "fish-tail" shape can be clearly observed in Fig.4.

Low Mach number flow over a circular bump

Here we present a numerical experiment concerning a low Mach number (=.1) flow over a flat wall with a circular (arch of $\pi/3$ and radius 2) bump. Here as well as in the previous case the presence of the bump is imitated through the appropriate boundary conditions. The grid is 200×200 points. The density contours of the steady-state solution are presented in Fig.5.

Multigrid algorithm

To illustrate the performance of the multigrid algorithm we consider here the well known test case of a shock reflecting from a flat wall. The multigrid algorithm involves five grids (levels): the finest consists of 129×33 points, the coarsest is 9×3 points.

The multigrid algorithm is based on the same two-dimensional scheme used with the lexicographic Gauss-Seidel relaxation. The restriction and prolongation procedures are the standard Full Weighting of the residuals and bilinear correction interpolation. The numerical solution to this problem obtained by the $2FMG - W(2,1)$

algorithm is presented on Fig.6(a). Fig.6(b) presents the numerical solution obtained using the same algorithm but performing three more cycles (five total) on the finest level.

Note that in this case the flow is aligned with the x-direction in a significant part of the domain. In this case the artificial viscosity in the cross-stream direction in the entropy and u-momentum equations vanishes. Therefore, no smoothing can be obtained in the y-direction in some components. A multigrid algorithm utilizing the time-stepping type relaxation can deal with such a situation only using the semi-coarsening technique. Our algorithm employs the Gauss-Seidel relaxation. Therefore, it offers a much simpler and more efficient treatment of this problem: relaxation with lexicographic ordering in the stream direction.

The rate of convergence observed in this test case as well as in other simple experiments concerning a variety of flow regimes is very close to .75.

DISCUSSION AND FUTURE WORK

Summary of the current work

A new two-dimensional high-resolution (at the steady-state) scheme for the compressible Euler equations was presented. It is triangle-based and can be formulated with the same degree of simplicity both on structured and unstructured grids. The main advantage of this scheme is that Gauss-Seidel relaxation can be applied directly to the resulting discrete equations. This allows construction of a simple and efficient multigrid steady-state solver.

A remarkable property of the constructed scheme is also its very compact stencil: it involves only the immediate neighbors of the point of interest.

A variety of flow regimes (supersonic, transonic and low Mach number flow) were considered in the numerical experiments to verify the quality of the solutions obtained by means of the new scheme and to demonstrate the efficiency of the multigrid algorithm.

Generalization of this scheme to three dimensional tetrahedral meshes is straightforward (see [9]).

Further improvement of the multigrid efficiency

The main obstacle preventing the further improvement of the multigrid efficiency is the following fact: for the hyperbolic problems the coarse grid correction is not sufficient for certain error components.

This difficulty was already addressed in the literature and some techniques to improve the multigrid efficiency were developed in [11]. Therefore, one possibility is to adapt these techniques for our case - compressible Euler equations.

REFERENCES

[1] Spekreijse, S., Multigrid solution of monotone second-order discretization of hyperbolic conservation laws, *Math. Comp.*, 49:135–155, 1987.

[2] Sidilkover, D., *Numerical solution to steady-state problems with discontinuities*, PhD thesis, The Weizmann Institute of Science, Rehovot, Israel, 1989.

[3] Sidilkover, D. and Brandt, A., Multigrid solution to steady-state 2D conservation laws, *SIAM J. Numer. Anal.*, 30:249–274, 1993.

[4] Deconinck, H., Struijs, R., and Roe, P. L., Fluctuation splitting for multidimensional convection problem: an alternative to finite volume and finite element methods, VKI Lecture Series 1990-3 on Computational Fluid Dynamics, Von Karman Institute, Brussels, Belgium, March 1991.

[5] Struijs, R., Deconinck, H., de Palma, P., Roe, P. L., and Powell, K. G., Progress on multidimensional upwind Euler solvers for unstructured grids, AIAA 91-1550, June 24-26 1991, Honolulu, Hawaii.

[6] Sidilkover, D. and Roe, P. L., Unification of some advection schemes in two dimensions, Report No. 95-10, ICASE, 1995, Submitted for publication.

[7] Roe, P. L., Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics, *J. Comp. Phys.*, 63:458–476, 1986.

[8] Deconinck, H., Hirsch, C., and Peuteman, J., Characteristic decomposition methods for the multidimensional Euler equations, in *Lecture Notes in Physics 264*, pp. 216–221, Springer, 1986.

[9] Sidilkover, D., A genuinely multidimensional upwind scheme and efficient multigrid solver for compressible Euler equations, Report No. 94-84, ICASE, 1994, Submitted for publication.

[10] Roe, P. L., Struijs, R., and Deconinck, H., A conservative linearization of the multidimensional Euler equations, To appear in J. Comp. Phys.

[11] Brandt, A. and Yavneh, I., Accelerated multigrid convergence and high Reynolds recirculating flows, *SIAM J. Sci. Statist. Comput.*, 14:607–626, 1993.

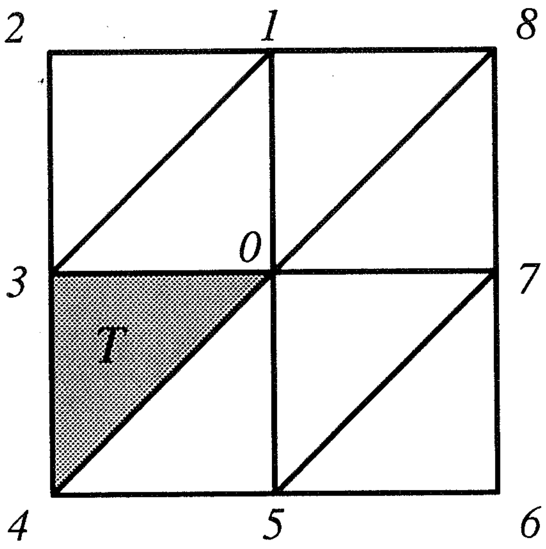


Figure 1: Triangulation.

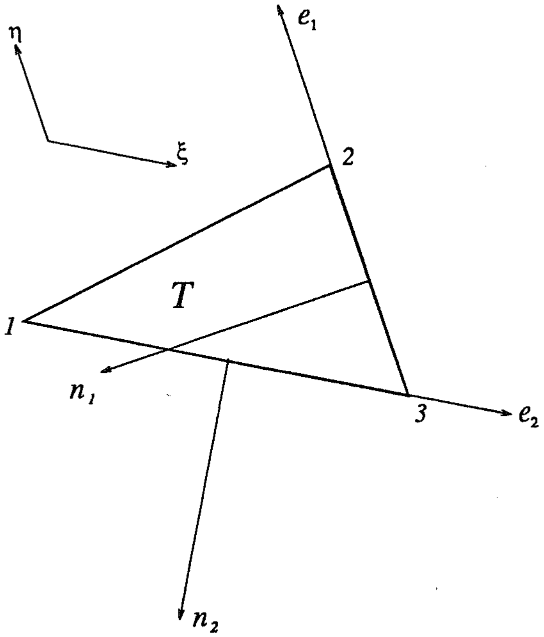


Figure 2: Triangle.

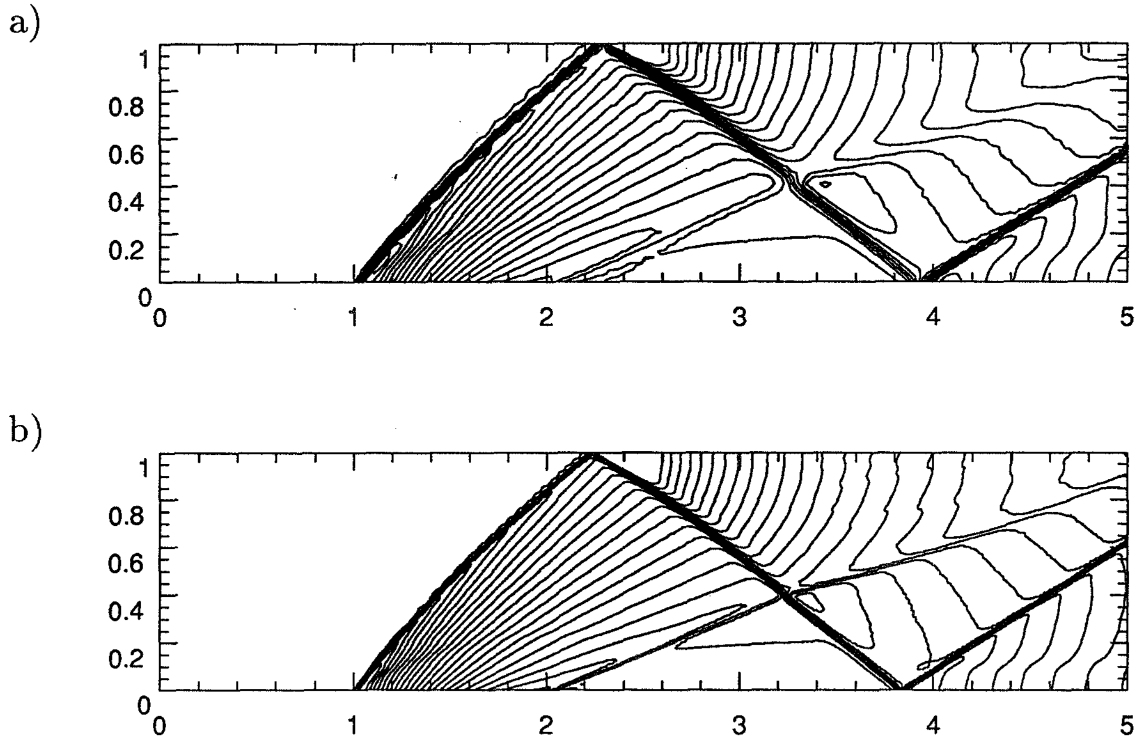


Figure 3: Supersonic flow in a channel over a circular bump: a) grid 200 × 40 pts.; b) the same, except the grid 400 × 80 pts.

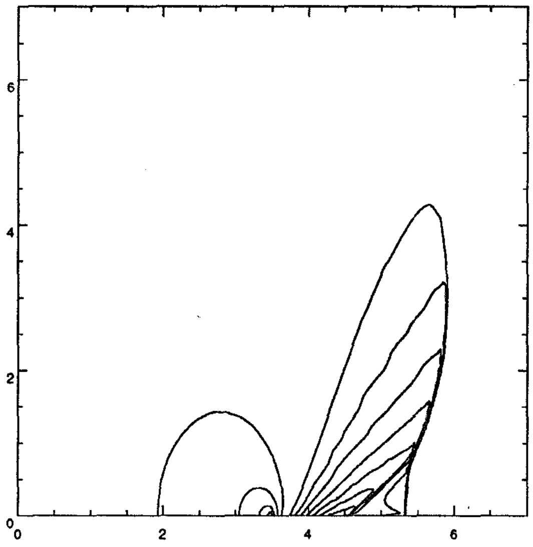


Figure 4: Transonic flow over a wall with a circular bump (free stream Mach= .9).

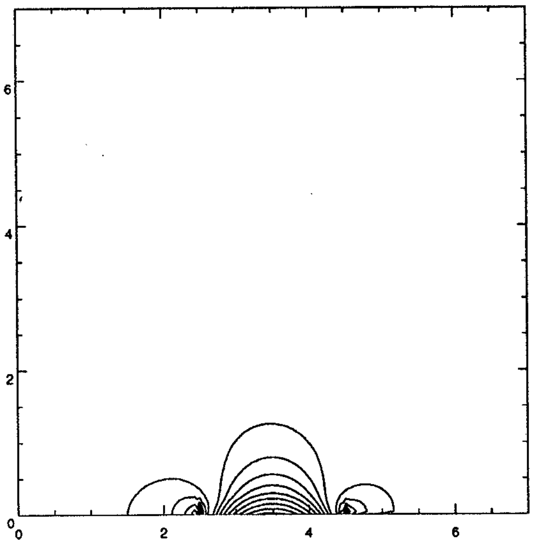


Figure 5: Low speed flow (Mach= .1) over a wall with a circular bump.

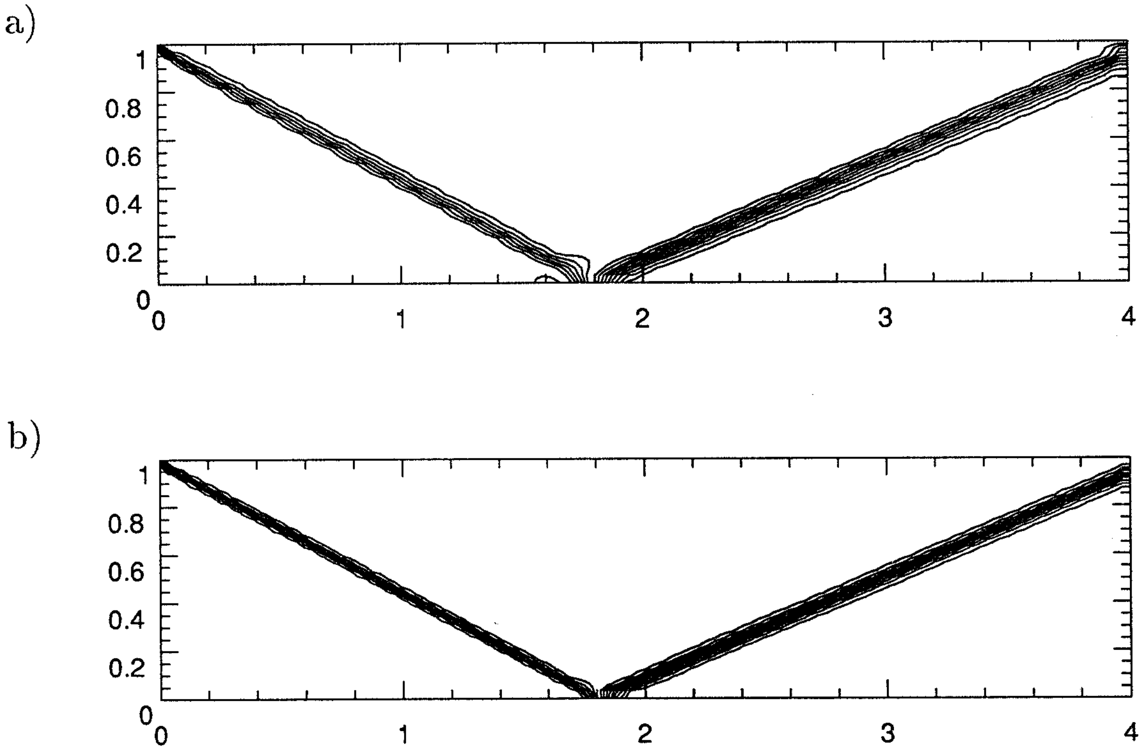


Figure 6: Performance of the multigrid algorithm, grid 129 × 33 pts.: a) solution obtained by $2FMG - W(2,1)$ algorithm; b) as previous after 3 more cycles on the finest grid.

Page intentionally left blank

ALGEBRAIC MULTIGRID BY SMOOTHED AGGREGATION FOR SECOND AND FOURTH ORDER ELLIPTIC PROBLEMS*

PETR VANĚK, JAN MANDEL, AND MARIAN BREZINA[†]

Summary. An algebraic multigrid algorithm is developed based on prolongations by smoothed aggregation. Coarse levels are generated automatically. Guidelines for the selection of method components are presented based on energy considerations. Efficiency of the resulting algorithm is demonstrated by computational results.

Key words. Algebraic multigrid, unstructured meshes, automatic coarsening, biharmonic equation

AMS(MOS) subject classifications. 65N55, 65F10

1. Introduction. Multigrid methods are very efficient iterative solvers for systems of algebraic equations arising from finite element and finite difference discretizations of elliptic boundary value problems. The main principle of multigrid methods is to complement the local exchange of information in point-wise iterative methods by a global one utilizing several related systems, called *coarse levels,* with a smaller number of variables. The coarse levels are often obtained as a hierarchy of discretizations with different characteristic meshsizes, but this requires that the discretization is controlled by the iterative method. To solve linear systems produced by existing finite element software, one needs to create an artificial hierarchy of coarse problems. The principal issue is then to obtain computational complexity and approximation properties similar to those for nested meshes, using only information in the matrix of the system and as little extra information as possible.

Such algebraic multigrid method that uses the system matrix only was developed by Ruge, et al. [10, 4, 11]. The prolongations were based on the matrix of the system by partial solution from given values at selected coarse points [1]. The coarse grid points were selected so that each point would be interpolated to via so-called strong connections.

Our approach is based on *smoothed aggregation* introduced recently by Vaněk [14, 13]. First the set of nodes is decomposed into small mutually disjoint subsets. A tentative piecewise constant interpolation (in the discrete sense) is then defined on those subsets as piecewise constant for second order problems, and piecewise linear for fourth order problems. The prolongation operator is then obtained by smoothing the output of the tentative prolongation and coarse level operators are defined variationally. Multigrid

* 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. The first author is visiting from University of West Bohemia, Americká 42, 306 14 Plzeň, Czech Republic

method based on such prolongations converges very fast for a wide range of problems including those with strongly anisotropic and discontinuous coefficients and, in addition, it has a remarkably low computational complexity since the typical coarsening ratio is about three in each dimension.

Almost optimal theoretical bounds for our method were given by the authors in [15] for second order problems and under natural assumptions on the coarse level hierarchy that tend to be satisfied by our coarsening algorithm, namely that the coarsening is by about the factor of three, and that the aggregates of the nodes are based on aggregated elements that form a reasonable mesh of macroelements. A bound on the energy of the coarse level basis functions was proved and used to verify the assumptions of the multilevel regularity-free approach of Bramble, Pasciak, Wang, and Xu [3]. The theory can be extended to fourth order problems once similar energy bounds are available for that case.

The part of this paper dealing with second order problems is based on [15]. The algorithm for fourth order problems is new. For more details and theory for the second order case, see [15].

For other multigrid approaches to the biharmonic equation, see [5, 9, 16, 8]. For a multigrid theory for the biharmonic equation with non-nested finite element spaces, see [2].

1.1. Basic Multigrid Algorithm. For reference, we state the basic multigrid algorithm for the solution of the system of linear algebraic equations $Ax = b$. First, a preprocessing stage creates full rank prolongation matrices P_l of size $n_l \times n_{l+1}$, $l = 1, \ldots, L-1$ by an automatic coarsening process described below. The coarse level matrices are defined by

$$A_1 = A, \qquad A_{l+1} = P_l^T A_l P_l, l = 1, \ldots, L-1.$$

The iterations then proceed as follows.

ALGORITHM 1 (BASIC MULTIGRID). *To solve the system $A_l x^l = b^l$, do:*

Pre-smoothing: *do ν_1 times $x^l \leftarrow S^l(x^l, b^l)$*

Coarse grid correction:

- *let $b^{l+1} \leftarrow P_l^T(b^l - A_l x^l)$*
- *If $l + 1 = L$, solve $A_{l+1} x^{l+1} = b^{l+1}$ by a direct method, otherwise apply γ iterations of this algorithm on level $l + 1$, starting with initial guess $x^{l+1} = 0$*
- *correct the solution on level l by $x^l \leftarrow x^l + P_l x^{l+1}$*

Post-smoothing: *do ν_2 times $x^l \leftarrow S^l(x^l, b^l)$.*

We use $\nu_1 = \nu_2 = \gamma = 1$ with the pre-smoothing iteration consisting of one forward iteration of the Gauss-Seidel followed by one iteration of backward SOR. The post-smoothing iteration consists of one forward SOR iteration followed by an iteration of backward Gauss-Seidel. The over-relaxation parameter used is 1.85 in both pre- and post-smoothing.

Each level is associated with basis functions $\{\varphi_i^l\}_{i=1}^{n_l}$. The basis functions on the finest level are given as finite element shape functions, while the coarse level basis functions are determined from the prolongations by

$$\begin{bmatrix} \varphi_1^{k+1} \\ \vdots \\ \varphi_{n_{k+1}}^{k+1} \end{bmatrix} = P_k^T \begin{bmatrix} \varphi_1^k \\ \vdots \\ \varphi_{n_k}^k \end{bmatrix} \qquad k = 1, \ldots, L-1.$$

2. Algebraic Multigrid for Second Order Problems.

Consider discretization by standard conforming linear finite elements of a second order elliptic variational problem

$$u \in V : \quad a(u,v) = f(v) \qquad \forall v \in V \tag{2.1}$$

where $V = H^1_{\Gamma_D}(\Omega)$ denotes the Sobolev space of H^1 functions vanishing on $\Gamma_D \subset \partial\Omega$, $\mu(\Gamma_D) > c\mu(\partial\Omega)$, Ω a domain in $\mathbb{R}^2$. The bilinear form

$$a(u,v) = \int_\Omega \sum_{i,j} a_{ij}\partial_i u \partial_j v \tag{2.2}$$

is assumed to be symmetric, V-elliptic, and bounded,

$$c_1\|u\|^2_{H^1(\Omega)} \leq a(u,u) \leq c_2\|u\|^2_{H^1(\Omega)}, \qquad \forall u \in V. \tag{2.3}$$

Moreover we assume that the finite element basis forms a decomposition of unity

$$\sum_{i=1}^{n_1} \varphi_i^1 = 1 \tag{2.4}$$

away from essential boundary conditions.

2.1. Construction of Prolongations for second order elliptic problems.

The prolongation operators are chosen to achieve low energy of coarse basis functions, leading to good theoretical estimates of the convergence of the iterations, as well as by sparsity considerations to achieve low computational complexity of the iterations. We are looking for prolongations that satisfy the following properties. First we specify the desired properties of the support of the coarse shape functions (or, equivalently, the allowed nonzeros of the prolongation matrices), and then the numerical values of the nonzero entries.

(AMG1) Coarse supports should follow strong couplings. We require that every two nodes in the support of a coarse basis function can be connected by a path of strong couplings. Two nodes i and j on level l are strongly coupled if $|a^l_{ij}|$ is relatively large compared with $\sqrt{|a^l_{ii}a^l_{jj}|}$. Essentially, we want to assure that the algorithm will provide the semi-coarsening in the case of solving of the anisotropic problem ([6], [12]). Algebraically, the anisotropy is reflected in the coefficients of the stiffness matrix in the sense that the neighboring nodes are strongly coupled in the direction of anisotropy.

(AMG2) Bounded intersection. Support of each basis function intersects a bounded number of supports of other basis functions on the same level only. The number of intersections does not depend on the level. This property guarantees sparsity of the resulting coarse-level matrices.

(AMG3) Decomposition of unity. Every coarse space V_l should represent the constant function exactly, aside from an essential boundary condition. This requirement is motivated by the need to bound locally the error of a coarse grid approximation $P_l v^{l+1}$ of a fine grid function u^l in terms of the energy $(u^l)^T A_l u^l$ and by the fact that the constant function has zero energy because of (2.2). Because of (2.4), this is equivalent to the requirement that the columns of each prolongation matrix form a decomposition of unity

$$\sum_{j=1}^{n_{l+1}} P_{ij} = 1, \qquad l = 1, \ldots, L-1,$$

for all rows i that do not correspond to degrees of freedom adjacent to an essential boundary condition. For generalizations, see Sections 3.1 and 3.3.

(AMG4) Small energy of coarse basis functions. We require that the energy of the coarse space basis functions be almost minimal in the sense that

$$\frac{a(\varphi_i^l, \varphi_i^l)}{||\varphi_i^l||^2_{L^2(\Omega)}} \leq C \inf_{u \in H_0^1(\text{supp}\varphi_i^l)} \frac{a(u,u)}{||u||^2_{L^2(\Omega)}}.$$

Note that in the case of uniformly V-elliptic problems the requirement above, together with bounded intersections of supports of basis functions (AMG2), assures the standard inverse inequality on each coarse space.

(AMG5) Uniform l^2 equivalence. Discrete l_2 norms on all spaces V_l should be uniformly equivalent up to diagonal scaling. The scaling may depend on the measure of the support of basis function and type of degree of freedom. For the algorithm described in this section, such uniform equivalence has been proved in [15].

We now construct prolongations P_l based on the matrix A_l. First we create a tentative piecewise constant prolongator satisfying all of the above properties except for the energy bound in (AMG4). This prolongator will then be smoothed to satisfy (AMG4), while preserving the other properties.

We start by specifying a disjoint decomposition of the set of nodes on level l. Every component of the decomposition on level l (so-called *aggregate*) gives rise to one degree of freedom on level $l+1$.

Motivated by the requirement (AMG1) above, for a given ε we define the *strongly-coupled neighborhood of node* i as

$$N_i^l(\varepsilon) = \{j : \; |a_{ij}| \geq \varepsilon\sqrt{a_{ii}a_{jj}}\} \cup \{i\} \tag{2.5}$$

Algorithm 2 (Aggregation). *Let the matrix A_l of order n_l and $\varepsilon \in [0,1)$ be given. Generate a disjoint covering $\{C_i^l\}_{i=1}^{n_{l+1}}$ of the set $\{1,\dots,n_l\}$ as follows.*

Initialization *Set $R = \{1,\dots,n_l\}$ and $j = 0$.*

Step 1 *Select disjoint strongly coupled neighborhoods as the initial attempted covering: If there exists a strongly coupled neighborhood $N_i^l(\varepsilon) \subset R$, set $j \leftarrow j+1$, $C_j^l \leftarrow N_i^l(\varepsilon)$, $R \leftarrow R \setminus C_j^l$. Repeat until R does not contain any strongly coupled neighborhood.*

Step 2 *Add each remaining $i \in R$ to one of the sets already selected to which it is strongly connected, if possible:*
Copy $\tilde{C}_k^l = C_k^l$, $k = 1,\dots,j$
If there exists $i \in R$ and k such that $N_i^l(\varepsilon) \cap \tilde{C}_k^l \neq \emptyset$ then set $C_k^l \leftarrow C_k^l \cup \{i\}$.
Repeat until no such i exists.

Step 3 *Make the remaining $i \in R$ into aggregates that consist of subsets of strongly coupled neighborhoods: If there exists $i \in R$, set $j \leftarrow j+1$ and $C_j^l = R \cap N_i^l(\varepsilon)$. Repeat until $R = \emptyset$.*

Define the tentative prolongation $\tilde{P}_l$ by the aggregates C_i^l:

$$(\tilde{P}_l)_{ij} = \begin{cases} 1 \text{ if } i \in C_j^l \\ 0 \text{ otherwise} \end{cases} \tag{2.6}$$

The piecewise constant prolongation $\tilde{P}_l$ will now be improved by a smoothing to get the final prolongation matrix P_l. We choose a simple Jacobi smoother, giving the prolongation matrix

$$P_l = (I - \omega D^{-1} A_l^F)\tilde{P}_l \tag{2.7}$$

where $A_l^F = (a_{ij}^F)$ is the *filtered matrix* given by

$$a_{ij}^F = \left\{ \begin{array}{cl} a_{ij} \text{ if } & j \in N_i^l(\varepsilon) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } i \neq j, \qquad a_{ii}^F = a_{ii} - \sum_{j=1, j\neq i}^{n_l} (a_{ij} - a_{ij}^F), \tag{2.8}$$

and D denotes the diagonal of A_l^F.

When applying Algorithm 2 to uniformly elliptic problems, one usually obtains the coarsening by about a factor of 3 in each dimension and the resulting coarse level matrix A_{l+1} tends to follow the nonzero pattern of the 9-point stencil. The filtration (2.8) has little or no effect in this case.

In the case of anisotropic problems, however, the application of the smoother with the unfiltered matrix would make the supports of basis functions overlap extensively in the direction of weak connections. Here the filtration prevents the undesired overlaps of the coarse space basis functions. By construction, A_l^F typically makes the nonzero pattern of A_{l+1} follow the 9-point stencil as in the uniformly V-elliptic case. It also assures that a constant remains the local kernel of A_l^F at every point where constant

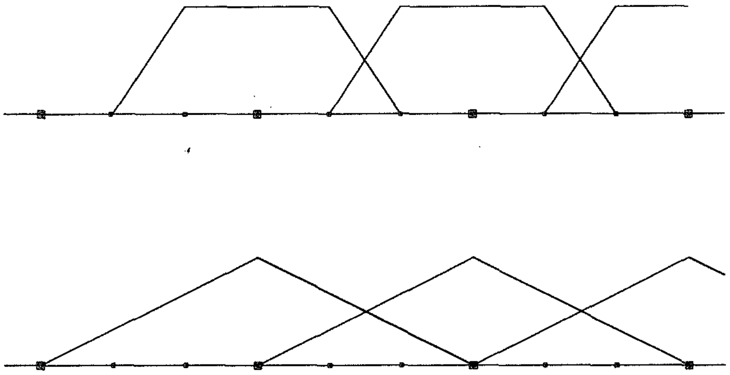


FIG. 2.1. *The basis functions given by aggregation and the corresponding smoothed basis for 1D Laplacian, using the smoother $I - 2/3D^{-1}A$.*

is the local kernel of A_l. Consequently, for problems without zero-order term the final prolongator P_l satisfies the decomposition of unity away from the essential boundary conditions.

Fig. 2.1 shows the 1D coarse basis functions resulting from prolongation by aggregation and the smoothed aggregation. Note that for the 1D Laplace operator and the choice of $\omega = 2/3$ in (2.7), the smoothed coarse space basis is exactly the one of $P1$-finite elements. Fig. 2.2 shows the typical aggregates obtained on an unstructured grid. The corresponding supports are formed by adding one belt of elements to the aggregates. The smoothing adds at most one more belt of adjacent elements.

We choose

$$\varepsilon = 0.08\,(\frac{1}{2})^{l-1}, \qquad \omega = \frac{2}{3}.$$

The theory for the above method can be found in [15].

3. Generalizations.

3.1. High order elements and unscaled problems.

The decomposition of unity (2.4) may be violated in practice. In such a case, in order to construct coarse spaces representing the constant function exactly, we need the representation of unity with respect to finite element basis of finest space V_1 as user input data. More specifically, we need the vector $\alpha \in \mathbb{R}^{n_1}$ satisfying

$$\sum_{i=1}^{n_1} \alpha_i \varphi_i^l = 1$$

away from essential boundary conditions.

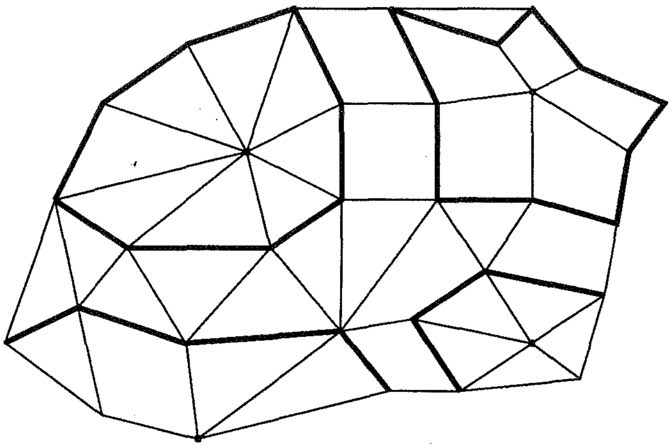


FIG. 2.2. *Typical 2D aggregates.*

The definition (2.6) of the auxiliary prolongators remains in place for all levels but level 1; we define $\tilde{P}_1$ as

$$\tilde{P}_{1_{ij}} = \begin{cases} \alpha_i \text{ if } i \in C_j^1 \\ 0 \text{ otherwise} \end{cases} \tag{3.1}$$

Thus, the unit constant function is represented by the vector $\alpha = (\alpha_i)_{i=1}^{n_1}$ on the finest level, while on levels 2 to L, the unit constant function is represented by vectors of all ones. The process can be easily generalized to the nonscalar case using the block approach described in Section 3.2. It was applied to the problem from Example No. 1 of Section 5 modified by scaling the basis functions randomly in the interval $[0.01, 1]$. The results are summed up in Example No. 5.

3.2. Vector problems. In the case of nonscalar problems, the coarsening algorithm as described in Section 2 is likely to produce aggregates of physically incompatible degrees of freedom causing deterioration of convergence. This phenomenon can, however, be overcome by using so-called block approach, which consists in replacing the scalar operations on the level of degree of freedom by their block counterparts on the level of node. Let n_d denote the number of degrees of freedom per node (assumed to be constant) and $df(i)$ be the list of degrees of freedom associated with the node i. The communication between the neighboring nodes k, l can now be expressed in the form of a matrix selection A_{kl} of order n_d

$$A_{kl} = A(df(k), df(l)). \tag{3.2}$$

The definition of strongly coupled neighborhood of node i (2.5) is now replaced by

$$N_i^l(\epsilon) = \{\, j : \; \|A_{ij}\| \geq \epsilon \sqrt{\|A_{ii}\| \|A_{jj}\|}\} \cup \{i\}, \tag{3.3}$$

where $||.||$ is a matrix norm. Further, in the definition of auxiliary prolongations (2.6), we replace the numbers 1 and 0 by identity and zero matrices of order n_d, respectively. The efficiency of this generalization is demonstrated by Experiments No. 1 and No. 5 in Section 5.

3.3. Absolute term. Consider now (2.1) modified by adding a positive absolute term

$$a(u,v) = \int_\Omega \sum_{i,j} a_{ij}\partial_i u \partial_j v + quv, \qquad q > 0.$$

In this case, the prolongation smoothers lose its constant-preserving property because the constant is no longer locally in the kernel of A_l. Fortunately, the presence of the absolute term improves the condition number of A_l, thus compensating for the loss of the preservation of a constant.

For large q, the absolute term also has the effect of boosting the diagonal dominance in certain (block) columns. The nodes corresponding to these columns are then treated by Algorithm 2 as isolated nodes, and the coarsening process may stall. Note that the same phenomenon may also result from certain treatments of the essential boundary conditions. This difficulty can easily be defeated by a simple modification. Removing these nodes from the set R in Algorithm 2 prevents the stalling. At the same time, it does not harm the convergence of the overall method, because the smoothers $\mathcal{S}^l$ are very efficient at approximating values in numerically isolated nodes.

4. Method for High order problems. For the elliptic problems of order $2K$, $K > 1$ requirements on prolongators have to be slightly stronger. Instead of decomposition of unity **(AMG3)** we now need the more general requirement.

(AMG3') Every coarse space V_l must represent polynomials of degrees up to $K-1$ exactly, away from the essential boundary conditions. As in the case of second order problems, this requirement is motivated by the need to control the coarse-grid approximation of $P_l v^{l+1}$ of u^l by energy $(u^l)^T A_l u_l$ and by the fact that norm and seminorm are equivalent on the factor space H^K modulo polynomials of degree of up to $K-1$.

Second, the small energy of coarse basis functions **(AMG4)** must be replaced by its straightforward generalization.

(AMG4') We require that the energy of coarse space basis functions be almost minimal in the sense that

$$\frac{a(\varphi_i^l, \varphi_i^l)}{||\varphi_i^l||^2_{L^2(\Omega)}} \le C \inf_{u \in H_0^K(\text{supp}\varphi_i^l)} \frac{a(u,u)}{||u||^2_{L^2(\Omega)}}.$$

Unfortunately, the construction of prolongators resulting in the coarse spaces satisfying **(AMG3')** for $K > 1$ is not possible without additional user input. In order to be able to approximate the polynomials with degrees of up to $K-1$ by coarse space functions exactly, we need their representation with respect to the finest level basis $\{\varphi_i^1\}_{i=1}^{n_1}$.

Finally, assumption **(AMG5)** may be satisfied with different scaling for each type of degree of freedom.

For the elliptic problem of order $2K$ on the domain $\Omega \subset \mathbb{R}^d$, we need vectors $p^{(0)}, p^{(ij)} \in \mathbb{R}^{n_1}$, $i = 1\dots, K-1$, $j = 1,\dots d$ satisfying

$$\sum_{k=1}^{n_1} p_k^{(0)}\varphi_k^1 = 1, \quad \sum_{k=1}^{n_1} p_k^{(ij)}\varphi_k^1 = x_j^i \tag{4.1}$$

away from the essential boundary conditions. For example, to solve the biharmonic equation in 2D, we need $p^{(0)}$, $p^{(11)}$, and $p^{(12)}$, the representations with respect to the fine-level basis of the planes $z = 1$, $z = x$, $z = y$, respectively.

The coarsening technique we are using is a natural generalization of the concept of smoothed aggregation described in Section 2.1. The aggregation step (2.6) can be viewed as a restriction of the unit vector to aggregates C_i^l, which gives rise to one degree of freedom on the level $l+1$ for each C_i^l. Here, tentative prolongators will be generated by restricting all the vectors p^0, p^{jk} to the aggregates C_i^1. Each aggregate will be represented by a set of degrees of freedom, where every degree of freedom corresponds to one of the vectors $p^{(0)}$, $p^{(jk)}$ (see Fig. 4.1). The shape of the basis functions derived from the nonconstant polynomials depends on the position of the aggregate. More specifically, being far away from the origin, basis functions derived from polynomials of higher degree contain a large low degree polynomial component which results in the violation of the uniform equivalence of discrete and continuous L_2–norms. This undesirable effect is suppressed by a local l_2 Gram-Schmidt orthogonalization process performed on each aggregate C_i^1 (see Fig. 4.2). Again, the resulting prolongator will be smoothed by the Jacobi smoother (see Fig. 4.3).

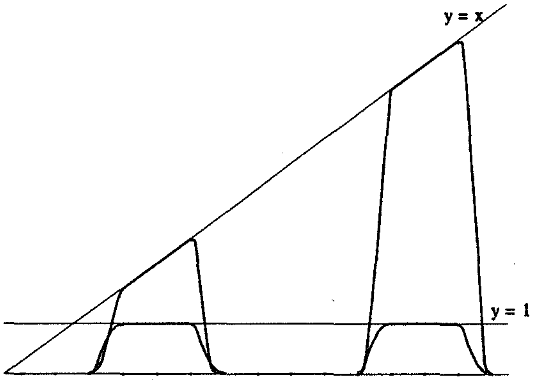


FIG. 4.1. *The coarse-space basis given by the restriction of p^0 and p^{11} onto aggregates of nodes.*

The following is a generalization of the algorithm of Section 2.1 to the case of problems of order $2K$, $K \geq 1$.

ALGORITHM 3 (COARSENING OF HIGH-ORDER PROBLEMS). *We assume the number of degrees of freedom per node on the finest level to be constant. Let n_1 be the number*

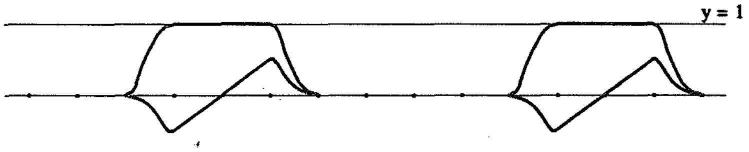


FIG. 4.2. *The coarse-space basis after l_2 Gram-Schmidt modification.*

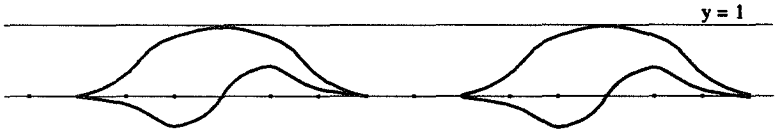


FIG. 4.3. *The final smoothed basis.*

of nodes on the finest level, and $df^1(i)$ denotes the list of degrees of freedom associated with the node i. We set $p^{1,(0)} = p^{(0)}, p^{1,(ij)} = p^{(ij)}, \quad i = 1, \ldots, K-1, \quad j = 1, \ldots, d$ (see (4.1)).

Step 1 - Decomposition. *Generate the disjoint covering $\{C_i^l\}_{i=1}^{n_{l+1}}$ of the set of nodes $\{1, \ldots, n_l\}$ using the Algorithm 2, where the strongly coupled neighborhood of i is defined by (3.3) and A_{ij} is the selection $A^l(df^l(i), df^l(j))$.*

Step 2 - Restriction. *For each aggregate C_i^l define the index set D_i^l of all degrees of freedom associated with nodes in C_i^l, i.e.*

$$D_i^l = \bigcup_{j \in C_i^l} df^l(j).$$

For every D_i^l generate auxiliary sparse vectors $v^{l,i,1}, \ldots, v^{l,i,n_p}$ by

$$v^{l,i,1} = p^{l,(0)}|_{D_i^l}, \; v^{l,i,2} = p^{l,(1,1)}|_{D_i^l}, \quad v^{l,i,3} = p^{l,(1,2)}|_{D_i^l}, \quad \ldots, v^{l,i,n_p} = p^{l,(K-1,d)}|_{D_i^l},$$

where $2K$ is the order of equation, d is the number of space variables ($\Omega \subset \mathbb{R}^d$), and $n_p = (K-1)d + 1$ is the number of the user supplied polynomials. $v|_I$ denotes the restriction of the vector to the index set in the sense that $(v|_I)_i = v_i$ if $i \in I$, zero otherwise.

Step 3 - Gram-Schmidt modification. *For each aggregate C_i^l update the set of associated sparse vectors generated in Step 2 by l_2 Gram-Schmidt orthogonalization process in the ordering $v^{l,i,1}, \; v^{l,i,2}, \ldots, \; v^{l,i,n_p}$ (i.e., vectors derived from low-degree polynomials are processed first). Note that the representation of the unity $v^{l,i,1}$ remains unchanged by the process.*

Step 4 - Building of auxiliary prolongators. *Generate the auxiliary prolongator $\tilde{P}_l$ whose $n_p(i-1) + j$-th column consists of the vector $v^{l,i,j}$ and create the corresponding coarse-level list of degrees of freedom associated with node i*

$$df^{l+1}(i) = \{n_p(i-1)+1, \; n_p(i-1)+2, \ldots, n_p i\}, \quad i = 1, \ldots, n_{l+1}.$$

Step 5 - Representation of polynomials on the coarse-level. *Generate vectors* $p^{l+1,(0)}, p^{l+1,(11)}, \ldots, p^{l+1,(K-1,d)}$ *satisfying*

$$p^{l,(0)} = \tilde{P}_l p^{l+1,(0)}, \quad p^{l,(11)} = \tilde{P}_l p^{l+1,(11)}, \ldots, \quad p^{l,(K-1,d)} = \tilde{P}_l p^{l+1,(K-1,d)}.$$

As $\{C_i^l\}$ *is a disjoint covering, the columns corresponding to different aggregates are* l_2*-orthogonal and consequently, the global Gram matrix given by columns of* $\tilde{P}_l$ *is a block matrix. Therefore,* $p^{l+1,(0)}$, $p^{l+1,(11)}, \ldots, p^{l+1,(K-1,d)}$ *can be computed by solving the local problems with Gram matrices generated by the columns of prolongator* $\tilde{P}_l$ *associated with* C_i^l

$$G_i^l = \{(v^{l,i,j}, v^{l,i,k})_{l_2}\}_{j,k=1}^{n_p}.$$

Step 6 - Final smoothing. *Improve the prolongator* $\tilde{P}_l$ *by smoothing step (2.7), (2.8), where scalar entries* a_{ij} *are replaced by blocks* $A_{ij} = A^l(df^l(i), df^l(j))$.

REMARK 4.1. Note that the final smoothed coarse basis functions resemble the standard shape functions for the Hermitean element with one degree of freedom for the value at the node and one degree of freedom for each derivative. This is true regardless of the choice of basis functions in the original problem (finest level), and makes an algebraic coarsening possible.

For the results of application of Algorithm 3, see Experiments 6 and 7 in Section 5.

REMARK 4.2. Efficient solution in the case of nonscalar problems of second order may also need the use of the coarsenig technique described in this section. For example, in the case of 3D elasticity, the energy norm is not equivalent to $(H^1)^3$-seminorm on the factorspace modulo constant in each field in the local sense, and consequently, the approximation property of the coarse space depends on the global constant of V-ellipticity, which can be very small if, for example, displacements are prescribed only on a rather small part of the boundary.

In order to eliminate the dependence of the convergence on boundary conditions, we need the prolongator to support the local kernels of the form, which will typically assure the desired local equivalence on the factorspace modulo kernel (i.e., local Korn's inequality on macroelements).

Thus, it is reasonable to build prolongators supporting the entire local kernels of the bilinear form instead of just a constant in each field. This can be achieved by supplying the representation of the basis vectors of the kernel in place of the vectors $p^{(0)}, p^{(11)} \ldots$. A similar technique that builds the coarse space from local generators of the nullspace is used in the so-called Balancing Domain Decomposition [7].

5. Numerical Experiments . The experiments in this section demonstrate the favorable behavior of the method. The code is available through anonymous ftp to `tiger.denver.colorado.edu` , directory /pub/faculty/pvanek. The experiments were performed on an IBM RS-6000/360 with 128 MBytes of memory.

experiment No.	rate of convergence	algebraic complexity	CPU time	real time
1	0.08	1.23	5s	5s
2	0.10	1.56	768s	7892s
3	0.21	1.14	134s	233s
4 a/b/c	0.11/0.10/0.10	1.65/1.65/1.65	85/85/85s	95/96/91s
5	0.09	1.24	13s	13s
6	0.26	1.37	64s	77s
7	0.31	1.48	114s	121s

TABLE 5.1
Results of numerical experiments.

The residual was measured in the l^2 norm. The iteration process was stopped once the relative residual became smaller than 10^{-5}. In all the experiments $V(1,1)$ cycle has been used. By algebraic complexity we mean the number of nonzero entries in the matrices on all the levels divided by the number of nonzeros in the matrix on finest level.

The rate of convergence is computed as an average reduction of l^2-norm of residual per iteration.

Results of experiments are summed up in Table 5.1. The description of testing problems follows.

EXPERIMENT NO. 1: Planar elasticity on unstructured mesh (Fig. 5.1). Poisson ratio 0.3, number of nodes 10610, number of degrees of freedom 21358. Boundary conditions : Dirichlet and Neumann.

EXPERIMENT NO. 2: Large anisotropic problem (5.1) with jumps in coefficients as in Fig. 5.2 and $q(x,y) = 0$. Number of nodes 10^6. The problem has been discretized on the regular square grid.

EXPERIMENT NO. 3: 3D problem (5.2) with random coefficients

$$w_{11} = \exp(rn_1),\ w_{22} = \exp(rn_2),\ w_{33} = \exp(rn_3),$$

where rn_i is a random number uniformly distributed in the interval $[\ln(10^{-2}), \ln(10^2)]$. Number of nodes 68921. The problem was discretized on the regular square grid.

$$\begin{gathered} -\frac{\partial}{\partial x}\, a(x,y)\frac{\partial u}{\partial x} - \frac{\partial}{\partial y}\, b(x,y)\frac{\partial u}{\partial y} + q(x,y)\, u = f(x,y) \ \text{ on } (0,1)\times(0,1), \\ u = 0 \ \text{ on } \partial\Omega. \end{gathered} \tag{5.1}$$

$$\begin{gathered} -\textstyle\sum_{i,j=1}^{3} \frac{\partial}{\partial x_i}(w_{ij}(x,y)\frac{\partial u}{x_j}) = f(x,y), \ \text{ on } (0,1)\times(0,1), \\ u = 0 \ \text{ on } \partial\Omega. \end{gathered} \tag{5.2}$$

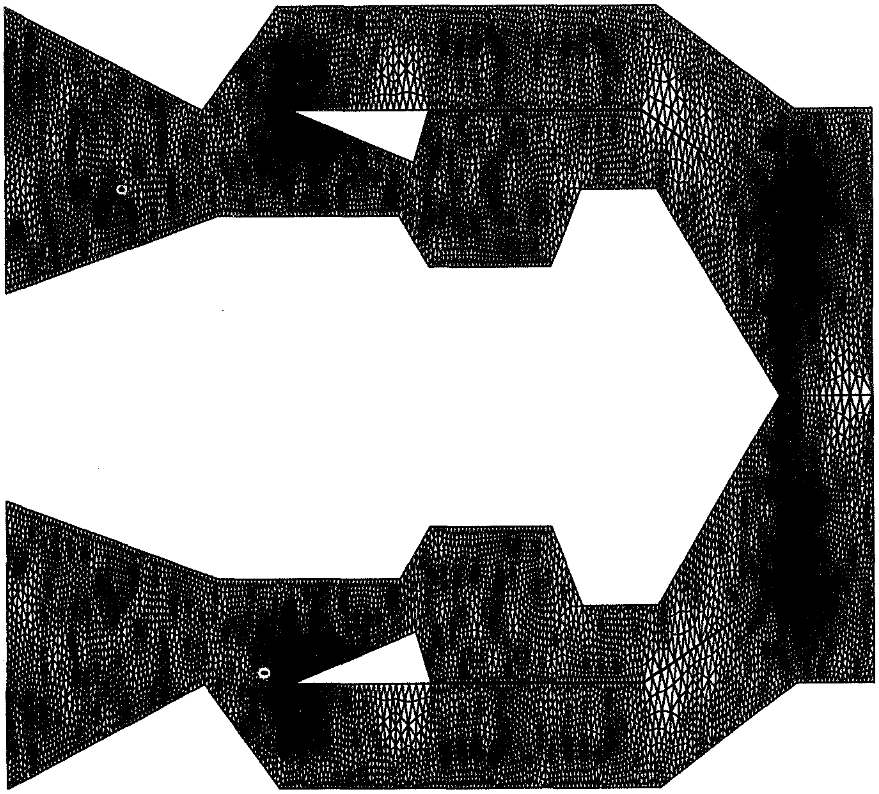


FIG. 5.1. *Mesh 1 (Courtesy of Charbel Farhat, Center for Aerospace Engineering, University of Colorado, Boulder).*

EXPERIMENT NO. 4: 2D anisotropic problem (5.1) with jumps in coefficients as in Fig. 5.2 and a) $q(x,y) = 0.1$, b) $q(x,y) = 1$, c) $q(x,y) = 10$. Number of nodes 160000. The problem was discretized on the regular square grid.

EXPERIMENT NO. 5: Planar elasticity on an unstructured mesh (Fig. 5.1) discretized by finite elements with randomly scaled basis. Poisson ratio 0.3, number of nodes 10610, number of degrees of freedom 21358. Boundary conditions : Dirichlet and Neumann.

EXPERIMENT NO. 6: Biharmonic problem discretized on the rectangular square grid. Number of degrees of freedom 48400. Boundary conditions: essential.

EXPERIMENT NO. 7: Fourth order problem (5.3) with coefficients given by (5.4) discretized on regular square grid. Number of degrees of freedom 48400. Boundary conditions: essential.

$$\frac{\partial^2}{\partial x^2}\left(a(x,y)\frac{\partial^2 u}{\partial x^2}\right) + \frac{\partial^2}{\partial x^2}\left(b(x,y)\frac{\partial^2 u}{\partial x^2}\right) = f(x,y) \text{ on } (0,1)\times(0,1) \qquad (5.3)$$

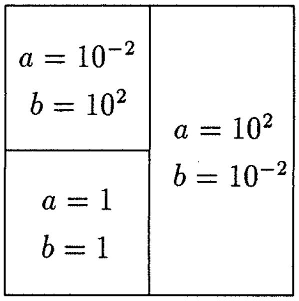


FIG. 5.2. *The coefficients $a(x,y)$, $b(x,y)$.*

$$a(x,y) = 1, \quad b(x,y) = e^{16xy} \tag{5.4}$$

The second order problems are discretized by $P1$ finite elements. The fourth order problems are discretized by a 27-point difference formula with Lagrangean degrees of freedom.

REFERENCES

[1] R. E. ALCOUFFE, A. BRANDT, J. E. DENDY, AND J. W. PAINTER, *The multi-grid methods for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 430–454.

[2] J. BRAMBLE AND X. ZHANG, *Multigrid methods for the biharmonic problem discretized by conforming C^1 finite elements on nonnested meshes.* Submitted.

[3] J. H. BRAMBLE, J. E. PASCIAK, J. WANG, AND J. XU, *Convergence estimates for multigrip algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.

[4] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge Univ. Press, Cambridge, 1984.

[5] S. C. BRENNER, *An optimal order nonconforming multigrid method for the biharmonic equation*, SIAM J. Numer. Anal., 26 (1989), pp. 1124–1138.

[6] J. E. DENDY, M. P. IDA, AND J. M. RUTLEDGE, *A semicoarsening multigrid algorithm for SIMD machines*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1460–1469.

[7] J. MANDEL, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.

[8] P. OSWALD, *Hierarchical conforming finite element methods for the biharmonic equation*, SIAM J. Numer. Anal., 29 (1992), pp. 1610–1625.

[9] P. PEISKER AND D. BRAESS, *A conjugate gradient method and a multigrid method for Morley's finite element approximation of the biharmonic equation*, Numer. Math., 50 (1987), pp. 567–586.

[10] J. W. RUGE, *Algebraic multigrid (AMG) for geodetic survey problems*, in Prelimary Proc. Internat. Multigrid Conference, Fort Collins, CO, 1983, Institute for Computational Studies at Colorado State University.

[11] J. W. RUGE AND K. STÜBEN, *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, in Multigrid Methods for Integral and Differential Equations,

D. J. Paddon and H. Holstein, eds., The Institute of Mathematics and Its Applications Conference Series, Clarendon Press, Oxford, 1985, pp. 169–212.

[12] R. A. Smith and A. Weiser, *Semicoarsening multigrid on a hypercube*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1314–1329.

[13] P. Vaněk, *Fast multigrid solver*. Applications of Mathematics, to appear.

[14] ——, *Acceleration of convergence of a two-level algorithm by smoothing transfer operator*, Applications of Mathematics, 37 (1992), pp. 265–274.

[15] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid on unstructured meshes*, Tech. Report 34, UCD/CCM, 1994.

[16] X. Zhang, *Multilevel Schwarz methods for the biharmonic Dirichlet problem*, SIAM J. Sci. Comput., 15 (1994), pp. 621–644.

Page intentionally left blank

Krylov Subspace and Multigrid Methods Applied to the Incompressible Navier-Stokes Equations

C. Vuik P. Wesseling S. Zeng
Faculty of Technical Mathematics and Informatics,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands,
e-mail: c.vuik@math.tudelft.nl, p.wesseling@math.tudelft.nl

Abstract

We consider numerical solution methods for the incompressible Navier-Stokes equations discretized by a finite volume method on staggered grids in general coordinates. We use Krylov subspace and multigrid methods as well as their combinations. Numerical experiments are carried out on a scalar and a vector computer. Robustness and efficiency of these methods are studied. It appears that good methods result from suitable combinations of GCR and multigrid methods.

1 Introduction

We compare various iterative methods for linear systems resulting from discretization of the time-dependent incompressible Navier-Stokes equations. Before discretization the physical domain is mapped onto a computational domain consisting of a number of rectangular blocks. In this paper we restrict ourselves to the one-block case and two space dimensions. For the space discretization we use finite volumes and a staggered grid. For the time discretization we use the Euler Backward finite difference scheme together with pressure correction.

Krylov subspace and multigrid methods are two types of promising iterative methods for the solution of large unsymmetric non-diagonally dominant linear systems of algebraic equations. These types of methods are much used to solve discretized Navier-Stokes equations. Our research using Krylov subspace methods is described in ([10], [11], [12]) and using multigrid methods is described in ([14], [15], [16], [4] - [6]).

As Krylov subspace method we choose the GMRESR method [9] (a combination of GCR [1] and GMRES [7]). For the multigrid method we use a Galerkin coarse grid approximation and two different smoothers.

Since many of the faster computers are vector computers, we also compare the vectorization properties of the different methods. Although probably in the near future parallel computers will supersede vector computers, the comparison will remain relevant because good vectorization properties imply in many cases good parallellization properties. Furthermore, vectorization aspects remain of interest because future high-performance parallel computing platforms will often contain vector processors. Finally, good vectorization normally implies good superscalar performance on many RISC processors. Note that GMRESR is easy to vectorize, since most of its arithmetic operations are vector updates, vector-vector and matrix-vector operations. Vector length becomes large as the grid is refined, which improves speed on vector computers. With respect to multigrid we have the following choices:

- use of a simple smoother, like point Jacobi, which is easily vectorized but not robust, or
- use of a more complicated smoother, like ILU, which is robust but harder to vectorize.

A disadvantage of multigrid methods is that the occurrence of vectors of short length is inevitable, since use of coarse grids is necessary. This diminishes multigrid efficiency on vector computers.

The foregoing observations on the advantages and disadvantages of the two types of methods suggest that combinations of them may be profitable. We compare the following methods:

Method 1: GMRESR with ILU preconditioning;
Method 2: Multigrid with Jacobi line smoothing;
Method 3: Multigrid with ILU smoothing;
Method 4: GCR with Method 2 as inner loop;
Method 5: GCR with Method 3 as inner loop.

In this paper, general boundary-fitted coordinates are used to compute flows in complicated geometries. In general coordinates, the incompressible Navier-Stokes equations are formulated in standard tensor notation as follows [8]:

$$\begin{aligned} \text{momentum equations} \quad \frac{\partial U^\alpha}{\partial t} + U^\beta U^\alpha_{,\beta} &= -g^{\alpha\beta} p_{,\beta} + Re^{-1}(g^{\beta\gamma}U^\alpha_{,\beta} + g^{\alpha\beta}U^\gamma_{,\beta})_{,\gamma}, \quad (1) \\ \text{continuity equation} \quad U^\alpha_{,\alpha} &= 0, \quad (2) \end{aligned}$$

where U^α is the contravariant representation of the velocity vector field, p the pressure, Re the Reynolds number, and $g^{\alpha\beta}$ the metric tensor. The range of Greek indices is

$\{1, 2\}$. We use a staggered grid arrangement and a lexicographic ordering of the grid points. Due to the use of virtual cells the number of u^1-, u^2-, and p-points is the same. Using finite volume discretization in space and the backward Euler method for time discretization, we obtain the following discrete systems at each time step (see [8] for details):

$$\frac{1}{\Delta t}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{n+1} - \frac{1}{\Delta t}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{n} = \begin{pmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \end{pmatrix}^{n+1} - \begin{pmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} & \mathbf{A}^{13} \\ \mathbf{A}^{21} & \mathbf{A}^{22} & \mathbf{A}^{23} \end{pmatrix}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{p} \end{pmatrix}^{n+1} \tag{3}$$

$$\begin{pmatrix} \mathbf{A}^{31} & \mathbf{A}^{32} \end{pmatrix}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{n+1} = 0, \tag{4}$$

where $\mathbf{u}^1$, $\mathbf{u}^2$ and $\mathbf{p}$ are algebraic vectors that approximate on the grid $\sqrt{g}\,U^1$ and $\sqrt{g}\,U^2$ and p, respectively, with $\sqrt{g}$ the Jacobian of the mapping, and $\mathbf{f}^1$ and $\mathbf{f}^2$ represent source terms. The nonlinear terms have been linearized with Newton's method. The linear operators $(\mathbf{A}^{31}\ \ \mathbf{A}^{32})$, resulting from discretization of the divergence operator in the continuity equation, and $\mathbf{A}^{13}$ and $\mathbf{A}^{23}$, resulting from discretization of the gradients of the pressure in the momentum equations, do not depend on time. The remaining operators are time-dependent.

Equations (3) and (4) are solved by the pressure correction method, as presented in [3], which consists of three steps. In the first step, the momentum equations are solved to give an intermediate value for the velocities, using the old pressure:

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{I} + \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \frac{1}{\Delta t}\mathbf{I} + \mathbf{A}^{22} \end{pmatrix}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{*} = \begin{pmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \end{pmatrix}^{n+1} + \frac{1}{\Delta t}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{n} - \begin{pmatrix} \mathbf{A}^{13} \\ \mathbf{A}^{23} \end{pmatrix}\mathbf{p}^n. \tag{5}$$

This equation system behaves like a discretization of a convection-diffusion equation. The main diagonal is enhanced by a contribution $1/\Delta t$ due to the time-derivative. Then the pressure equation, which is derived from the momentum equation (3) and the continuity equation (4), is solved to give the difference $\mathbf{p}^{n+1} - \mathbf{p}^n$:

$$\begin{pmatrix} \mathbf{A}^{31} & \mathbf{A}^{32} \end{pmatrix}\begin{pmatrix} \mathbf{A}^{13} \\ \mathbf{A}^{23} \end{pmatrix}(\mathbf{p}^{n+1} - \mathbf{p}^n) = -\frac{1}{\Delta t}\begin{pmatrix} \mathbf{A}^{31} & \mathbf{A}^{32} \end{pmatrix}\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{*} \tag{6}$$

The coefficient matrix of $\mathbf{p}^{n+1} - \mathbf{p}^n$ does not change with time, and resembles a discretization of the Laplacian operator (in general coordinates), but is not symmetric. Finally, the velocities at time step $n+1$ are computed by means of

$$\begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{n+1} = \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix}^{*} + \Delta t\begin{pmatrix} \mathbf{A}^{13} \\ \mathbf{A}^{23} \end{pmatrix}(\mathbf{p}^{n+1} - \mathbf{p}^n). \tag{7}$$

In the next section we describe the iterative methods used for the solution of (5) and (6).

2 Solution Methods

In this section the iterative methods to be tested are described. The GMRESR method combined with ILU type preconditioners is given in Subsection 2.1. This is a summary of the methods described in [12]. In Subsections 2.2.1 and 2.2.2, the multigrid methods using an alternating Jacobi line smoothing and an ILU smoothing are presented. New methods, consisting of combinations of GMRESR and multigrid, are proposed in Subsection 2.2.3.

2.1 Method 1: GMRESR with ILU preconditioning

In Section 1 we have seen that there are two types of linear systems to be solved: the momentum equations and the pressure equation. Each has its own characteristic properties. We use GMRESR for both but with different preconditioners. The GMRESR method is defined in [9], successfully applied to the Navier-Stokes equations in [11], and analysed further in [10]. The GMRESR algorithm can be formulated as follows:

```
Algorithm GMRESR
  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,\ k = -1$
  while $\|\mathbf{r}_{k+1}\|/\|\mathbf{r}_0\| > tol$ do
    $k := k + 1$
    apply one iteration of GMRES($m$) to $\mathbf{A}\mathbf{y}_k = \mathbf{r}_k$ and
    denote the result by $\mathbf{u}_k^{(0)}$
    $\mathbf{c}_k^{(0)} = \mathbf{A}\mathbf{u}_k^{(0)}$
    for $i = 0, 1, \cdots, k-1$ do
      $\alpha_i = \mathbf{c}_i^T \mathbf{c}_k^{(i)}$
      $\mathbf{c}_k^{(i+1)} = \mathbf{c}_k^{(i)} - \alpha_i \mathbf{c}_i;\ \mathbf{u}_k^{(i+1)} = \mathbf{u}_k^{(i)} - \alpha_i \mathbf{u}_i$
    end do
    $\mathbf{c}_k = \mathbf{c}_k^{(k)}/\|\mathbf{c}_k^{(k)}\|_2;\ \mathbf{u}_k = \mathbf{u}_k^{(k)}/\|\mathbf{c}_k^{(k)}\|_2$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k \mathbf{c}_k^T \mathbf{r}_k;\ \mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{c}_k \mathbf{c}_k^T \mathbf{r}_k$
  end while
```

GMRESR consists of a GCR outer loop and a GMRES inner loop. In every outer iteration, m iterations are used in the GMRES inner loop. Only in the final outer iteration it is possible to do less than m inner iterations (see [9]). In this paper the GMRESR algorithm is used with the '*min alfa*' truncation strategy (see [10]). A truncation strategy is necessary to restrict the required memory. Truncation means the following: choose the number (*ntrunc*) of search directions ($\mathbf{u}_k$) that may be kept in memory. If the number of iterations becomes larger than *ntrunc*, a search direction $\mathbf{u}_j$ and its companion $\mathbf{c}_j (= \mathbf{A}\mathbf{u}_j)$ are overwritten by the new search direction $\mathbf{u}_{k+1}$ and $\mathbf{c}_{k+1}$. The *min alfa* truncation strategy is a method to decide which search direction should be discarded by the following criterion: find j such that $\alpha_j = \mathbf{c}_j^T \mathbf{c}_{k+1}^{(j)}$

satisfies the following equation:

$$|\alpha_j| = \min_{0\leq i\leq ntrunc} |\alpha_i|. \tag{8}$$

To obtain an efficient solver, GMRESR is combined with a preconditioner. For the pressure equation we use the classic incomplete LU decomposition (all fill-in is neglected). For the details of this preconditioner and the combination with GMRESR we refer to [12]. We use an ILUD preconditioning for the momentum equations. In this type of preconditioning the off-diagonal parts of L and U are the same as that of the given matrix and only the diagonal is adapted. In all the numerical experiments given in Section 3, we use the GMRESR(5) method (so m = 5).

2.2 Multigrid methods

In this paper we use multigrid methods consisting of the F-cycle with one pre- and one post-smoothing. In Subsection 2.2.1 the coarse grid operators are defined. The two smoothing operators used are given in Subsection 2.2.2, corresponding to Methods 2 and 3. In Subsection 2.2.3 the combined methods are given.

2.2.1 Formulation of coarse grid operators

Coarse grid operators are formulated by means of Galerkin coarse grid approximation [13]. For brevity, we write equations (5) and (6) as

$$\begin{pmatrix} \mathbf{A}^{11} & \mathbf{A}^{12} \\ \mathbf{A}^{21} & \mathbf{A}^{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \end{pmatrix}, \tag{9}$$

$$\mathbf{A}^{33}\mathbf{p} = \mathbf{f}^3. \tag{10}$$

Let l be the grid index, with $l = 1$ indicating the coarsest grid. Galerkin coarse grid approximation is carried out from grid $l+1$ to grid l as follows:
momentum equations

$$\begin{pmatrix} \mathbf{A}^{11(l)} & \mathbf{A}^{12(l)} \\ \mathbf{A}^{21(l)} & \mathbf{A}^{22(l)} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^1\mathbf{A}^{11(l+1)}\mathbf{P}^1 & \mathbf{R}^1\mathbf{A}^{12(l+1)}\mathbf{P}^2 \\ \mathbf{R}^2\mathbf{A}^{21(l+1)}\mathbf{P}^1 & \mathbf{R}^2\mathbf{A}^{22(l+1)}\mathbf{P}^2 \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}^{1(l)} \\ \mathbf{f}^{2(l)} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^1\mathbf{r}^{1(l+1)} \\ \mathbf{R}^2\mathbf{r}^{2(l+1)} \end{pmatrix} \tag{11}$$

and pressure equation

$$\mathbf{A}^{33(l)} = \mathbf{R}^3\mathbf{A}^{33(l+1)}\mathbf{P}^3, \quad \mathbf{f}^{3(l)} = \mathbf{R}^3\mathbf{r}^{3(l+1)}. \tag{12}$$

The **r**'s are the residuals, for example, $\mathbf{r}^3 = \mathbf{f}^3 - \mathbf{A}^{33}\mathbf{p}$. Here the **R**'s and **P**'s are restriction operators and prolongation operators, which are described below.

Standard cell-centered coarsening is used: a cell on the next coarse grid is formed by taking the union of four fine grid cells. The restriction operators $\mathbf{R}^1$ and $\mathbf{R}^2$ are for the momentum equations and $\mathbf{R}^3$ for the pressure equation. The prolongation operators $\mathbf{P}^1$, $\mathbf{P}^2$ and $\mathbf{P}^3$ are applied to $\mathbf{u}^1$, $\mathbf{u}^2$ and $\mathbf{p}$, respectively. The prolongation used for the coarse grid corrections is the same as in Galerkin coarse grid approximation.

The operators $\mathbf{R}^1$ and $\mathbf{R}^2$ use so-called hybrid interpolation, which, for example for $\mathbf{R}^1$, is obtained by using the adjoint of linear interpolation for $\mathbf{u}^1$ in direction 1 but the adjoint of piecewise constant interpolation in direction 2. Operator $\mathbf{R}^3$ is simply the adjoint of piecewise constant interpolation. Operators $\mathbf{R}^1$ and $\mathbf{R}^3$ are given by

$$\left[\mathbf{R}^1\right] = \frac{1}{2}\begin{bmatrix} we & 2 & \underline{we} \\ we & 2 & we \end{bmatrix}, \quad \left[\mathbf{R}^3\right] = \frac{1}{2}\begin{bmatrix} 1 & \underline{1} \\ 1 & 1 \end{bmatrix}, \tag{13}$$

where $w = 0$ when the 'west' points are on or outside of the 'west' boundary and $w = 1$ elsewhere, and similarly for s, e and n. $\mathbf{R}^2$ is similar to $\mathbf{R}^1$. The elements with an underscore correspond to the fine grid point $2k$ when restriction results in a function value in the coarse grid point k. The prolongation operators $\mathbf{P}^1$, $\mathbf{P}^2$ and $\mathbf{P}^3$ employ bilinear interpolation. The adjoints $\mathbf{P}^{1*}$ and $\mathbf{P}^{3*}$ of $\mathbf{P}^1$ and $\mathbf{P}^3$ are given by:

$$\begin{aligned} \left[\mathbf{P}^{1*}\right] &= \frac{1}{8}\begin{bmatrix} nw & 2n & ne \\ (4-n)w & 2(4-n) & \underline{(4-n)e} \\ (4-s)w & 2(4-s) & (4-s)e \\ sw & 2s & se \end{bmatrix}, \qquad (14) \\ \left[\mathbf{P}^{3*}\right] &= \frac{1}{16}\begin{bmatrix} nw & n(4-w) & n(4-e) & ne \\ (4-n)w & 16-4(n+w)+nw & \underline{16-4(n+e)+ne} & (4-n)e \\ (4-s)w & 16-4(s+w)+sw & 16-4(s+e)+se & (4-s)e \\ sw & s(4-w) & s(4-e) & se \end{bmatrix}, \end{aligned}$$

and $\mathbf{P}^{2*}$ is similar to $\mathbf{P}^{1*}$. For a more detailed exposition of these transfer operators, see [13] and [16].

2.2.2 The smoothing operators

In this subsection we describe the smoothers which are used in the multigrid method: Jacobi smoothing and ILU smoothing. The reason for this choice is that Jacobi smoothing has good vectorization (parallellization) properties but is not robust, whereas the ILU smoothing is robust but not easily vectorized.

Method 2: Multigrid with Jacobi smoothing
Our Jacobi smoothing method consists of one horizontal Jacobi line iteration followed by one vertical Jacobi line iteration. The momentum equations are smoothed in a decoupled way, i.e., the two momentum equations are smoothed successively. In a horizontal smoothing iteration, mutually independent tridiagonal systems have to be

solved: $M_j \delta x_j = r_j$ for a horizontal line j. The three non-zero elements at row i in M_j are denoted by $l_{i,j}, \hat{d}_{i,j}$, and $u_{i,j}$. The matrix M_j is factorised into:

$$M_j = (L_j + D_j) D_j^{-1} (D_j + U_j) \tag{15}$$

where L_j and U_j have only one non-zero diagonal below and above the main diagonal, equal to $l_{i,j}$ and $u_{i,j}$ and D_j is a diagonal matrix. Comparable formulae are used in a vertical smoothing iteration. Variables are updated after each horizontal and after each vertical step with a fixed underrelaxation factor $w = 0.7$.

Method 3: Multigrid with ILU smoothing
Suppose that the equation to be smoothed is denoted by

$$\mathbf{Ax} = \mathbf{b}. \tag{16}$$

A smoothing iteration is given by

$$\delta\mathbf{x} = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{Ax}), \mathbf{x} := \mathbf{x} + \omega\delta\mathbf{x} \tag{17}$$

with $\omega = 0.8$ fixed. For the ILU smoothing we choose $\mathbf{M} = (\mathbf{L} + \mathbf{D})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})$, where $\mathbf{L}$ and $\mathbf{U}$ are strictly lower and upper triangular matrices, and $\mathbf{D}$ a diagonal matrix. Matrices $\mathbf{L}$ and $\mathbf{U}$ have non-zero entries in the positions corresponding to the standard 9-point stencil pattern and are chosen such that the elements of $\mathbf{M}$ belonging to the 9-point pattern are equal to the corresponding elements of $\mathbf{A}$. The momentum equations are smoothed in the same decoupled manner as in Method 2. Again, factorization takes place only at the beginning of multigrid iterations for a time step, and $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ are kept until the next time step.

2.2.3 The combined methods

The methods presented below are very flexible. In many other combinations of Krylov subspace and multigrid methods, the inner loop procedure must be the same for every outer loop iteration. In these methods this is not necessary, so in different outer iterations one may use different inner loops, for instance a mix of GMRES and multigrid, or a different number of iterations with multigrid or multigrid with different smoothers, etc. The methods are based on the GMRESR idea where we use a GCR outer loop and a GMRES inner loop. The algorithms for the new methods are given below and only differ in the construction of the new search directions.

Method 4: GCR with Method 2 as inner loop
This method is obtained by replacing GMRES(m) in the inner loop of Method 1, by Method 2.

Method 5: GCR with Method 3 as inner loop
This method is obtained by replacing GMRES(m) in the inner loop of Method 1, by Method 3.

3 Numerical Experiments

3.1 Test Problems

We consider four test problems: an oblique driven cavity problem, an L-shaped driven cavity problem, a backward facing step problem [2], and a 90^o bend problem [11]. The grids used for these problems are shown in Figure 1. We study these problems for various time steps and grid sizes. Furthermore for every problem two values of the Reynolds number are used. For the driven cavity problems we take $Re_{low} = 1$ and $Re_{high} = 1000$, in the backward facing step problem $Re_{low} = 50$ and $Re_{high} = 150$, whereas in the bend problem $Re_{low} = 500$ and $Re_{high} = 1000$. The number of time steps is fixed at 40. This number is a rather arbitrary choice, because our purpose here is not to solve problems until steady state, but to investigate the performance (efficiency and robustness) of solution methods. Based on numerical experiments, the following stop criterion is chosen: the iterative solution of the systems at each time step is terminated if the ratio of the norm $\|\mathbf{r}\|$ of the residual to the norm $\|\mathbf{r}_0\|$ of the residual at the beginning of the present time step satisfies $\|\mathbf{r}\|/\|\mathbf{r}_0\| < tol$, with $tol = 10^{-4}$ for the momentum equations and $tol = 10^{-6}$ for the pressure equation. In Subsection 3.2 experiments on a scalar computer are described whereas Subsection 3.3 contains the results on a vector computer.

3.2 Experiments on a scalar computer

In this subsection we present numerical experiments on an HP 735 computer. We have run all methods described in Section 2 for the test problems given in Subsection 3.1. For brevity, here we only present a representative subset of the results. In Subsection 3.2.1 the momentum equations are considered, whereas in Subsection 3.2.2 we show results for the pressure equation.

3.2.1 The momentum equations

The properties of the linear systems originating from the discretized momentum equations that influence the iterative solvers depend on: the size of the time step, the Reynolds number, the grid size, and the shape of the space domain. Below, the influence of these parameters is considered in more detail. In the first part we restrict ourselves to the oblique driven cavity problem, only in the final part results are given for all test problems. The reason for this is that the results for the other problems are comparable with those of the oblique driven cavity problem.

Dependence on Δt, the Reynolds number and the grid size

In Table 1 we give some measurements concerning Method 1 and Method 3 applied

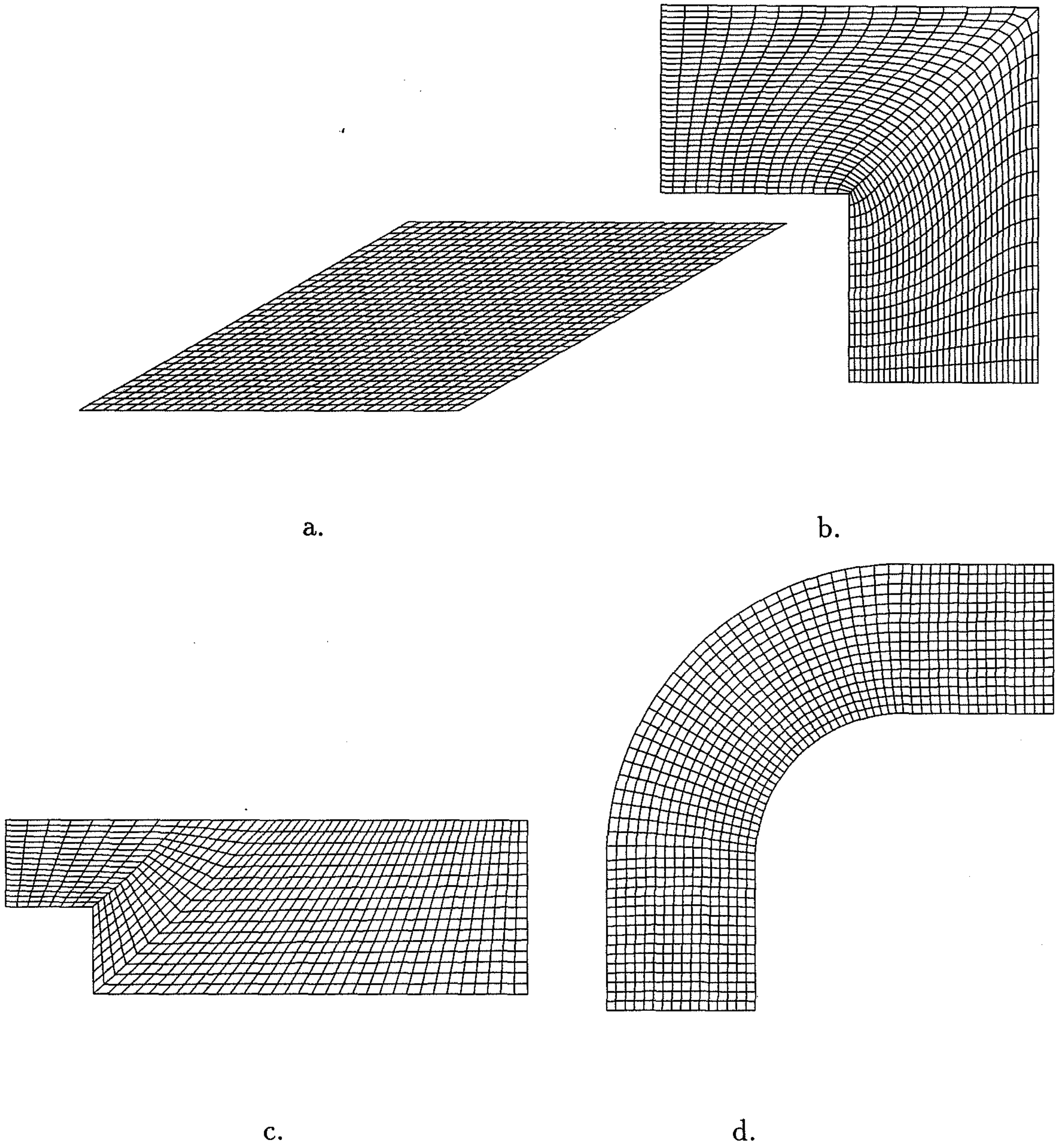


Figure 1: Grids for the four test problems: a. The oblique driven cavity problem (32 × 32); b. The L-shaped driven cavity problem (32 × 32); c. The backward facing step problem (48 × 16); d. The 90° bend problem (16 × 64).

		$Re = 1$				$Re = 1000$			
Grid	Δt	t_t	t_v, t_p	k_v, k_p	ρ_v, ρ_p	t_t	t_v, t_p	k_v, k_p	ρ_v, ρ_p
Method 1									
32	.0625	19	7, 7	5, 9		13	1, 7	1, 8	
×	.125	19	7, 7	5, 8		14	2, 7	2, 8	
32	.25	19	8, 7	5, 8		15	3, 7	3, 8	
64	.0625	151	74, 57	8,13		90	14, 56	2, 12	
×	.125	158	81, 57	9,12		93	18, 55	3, 11	
64	.25	162	86, 57	10,13		104	28, 56	4, 12	
128	.0625	1501	774,642	14,22		830	97,648	2, 22	
×	.125	1617	879,653	18,22		870	132,652	4, 22	
128	.25	1655	917,653	20,23		951	213,653	6, 23	
Method 3									
32	.0625	74	26, 35	4,14	.246,.371	63	20, 30	3, 12	.0871,.366
×	.125	74	27, 35	4,14	.240,.373	66	24, 30	4, 11	.142 ,.313
32	.25	74	27, 34	4,14	.226,.372	72	29, 30	6, 11	.250 ,.346
64	.0625	257	97,122	4,14	.229,.370	224	76,110	3, 12	.0933,.351
×	.125	255	98,120	4,14	.215,.371	240	91,111	4, 11	.138 ,.366
64	.25	252	97,117	4,13	.200,.370	240	93,109	4, 11	.214 ,.357
128	.0625	1073	424,499	4,13	.203,.370	1015	395,470	4, 10	.163 ,.354
×	.125	1058	425,484	4,13	.194,.370	1056	410,496	4, 11	.179 ,.338
128	.25	1045	425,470	4,12	.190,.368	1099	417,532	4, 12	.191 ,.358

Table 1: The oblique driven cavity problem on the HP: the total CPU time t_t, the CPU times t_v and t_p for the solution of the momentum equations and the pressure equation, respectively, the numbers of iterations k_v and k_p in the final time step, and the reduction factors ρ_v and ρ_p of the multigrid algorithm in the last iteration in the final time step.

to the oblique driven cavity problem. The behaviour of the other methods is comparable to Method 3. We observe that the number of iterations of Method 3 is more or less independent to the various choices of Δt, Re, or the grid size.

Now, we consider the dependence of Method 1 (GMRESR) for the various choices. The main diagonal of the momentum matrix is enhanced by a contribution $1/\Delta t$ due to the time derivative. So for small Δt the matrix is diagonal dominant. It appears from Table 1 that the number of iterations of the GMRESR method grows, if Δt increases. Comparing the results for the two Reynolds numbers, it appears that GMRESR converges much faster for $Re = 1000$ than for $Re = 1$. Finally, as expected, the number of GMRESR iterations increases for increasing grid size.

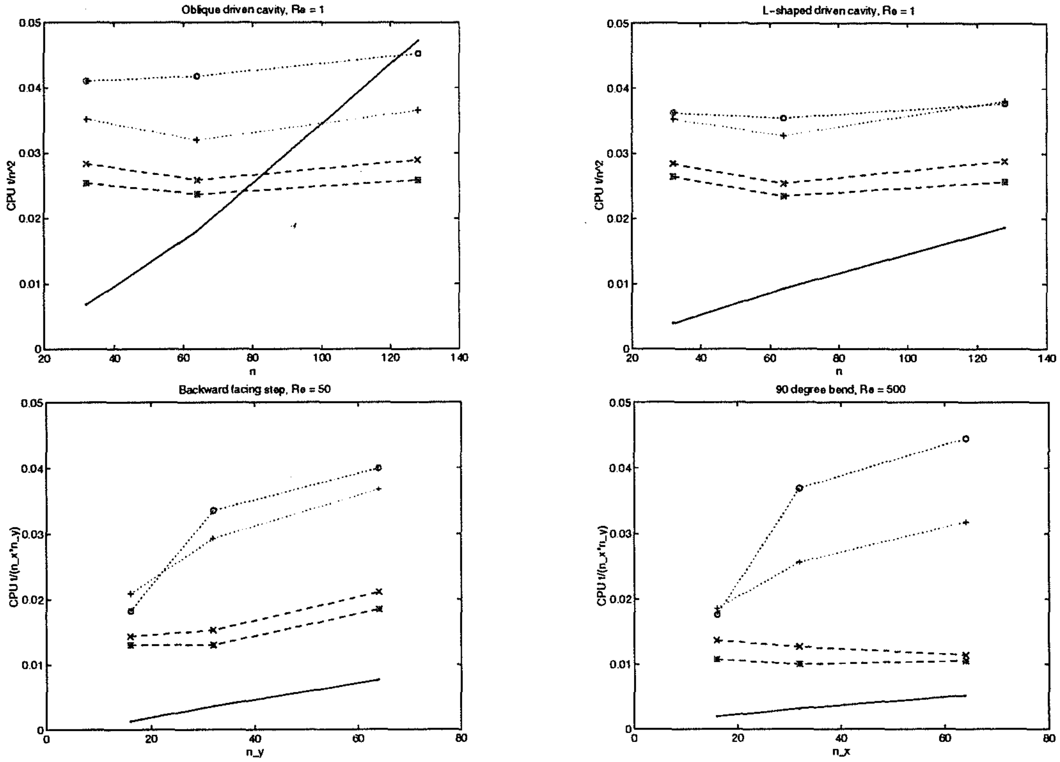


Figure 2: CPU times per grid point on the HP for the momentum equation during 40 time steps, for Re_{low} and $\Delta t = 0.0625$.

Problem dependence and comparison

For a comparison of the various methods on the four test problems we plot the CPU time on an HP 735 per grid point for 40 time steps against the grid size. In these figures we use the following symbols:

Method 1:	solid lines and point marks,
Method 2:	dotted lines and circles,
Method 3:	dashed lines and stars,
Method 4:	dotted lines and plus marks,
Method 5:	dashed lines and x-marks.

Where no symbols are shown they are off-scale. For Re_{low} the results are given in Figure 2 and for Re_{high} the results are given in Figure 3.

First we discuss the combination of GCR and multigrid. From Figures 2 and 3 it appears that the GCR acceleration of the Jacobi smoothed multigrid is better than multigrid itself. If the smoother is sufficiently powerful, as for instance for Method 3. where we use an ILU smoother, then the combination of GCR and multigrid gives a slightly worse performance. In these cases, the number of iterations is the same but the CPU time increases somewhat due to the GCR overhead.

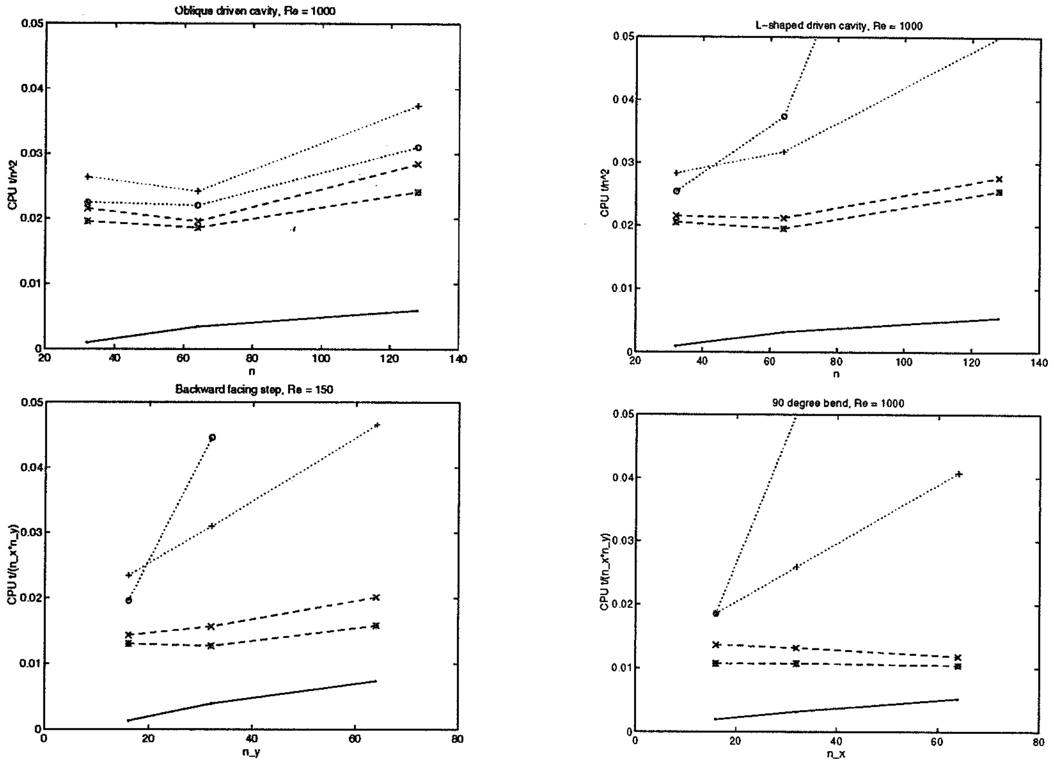


Figure 3: CPU times per grid point on the HP for the momentum equations during 40 times steps for Re_{high} and $\Delta t = 0.0625$.

Secondly, we compare Method 1 with the best multigrid method: Method 3. It appears that for Method 3 the CPU time per grid point is independent of the grid size and the Reynolds number. For Method 1 there is more variation: the CPU time increases for a larger grid size and a smaller Reynolds number. For a large Reynolds number Method 1 is much faster than Method 3. For the driven cavity problems and a small Reynolds number, Method 1 is more efficient for medium grid sizes, whereas Method 3 is the best method for large grid sizes. For the oblique driven cavity problem the break-even point is in the range [64, 128] and for the L-shaped driven cavity problem the break-even point is in the range [128, 256].

Finally we discuss robustness. Methods 1, 3 and 5 are equally robust. For most problems they work well. Only for the 90^o bend problem there are some failure cases (not shown here) when Δt is large and Re large. The least robust method is Method 2; it suffers from convergence problems when either the grid is refined or Δt is large for some problems. But when it is combined with GCR, resulting in Method 4, robustness is improved very much. Sometimes when Method 2 fails to work, Method 4 still works rather satisfactorily. However, Method 4 falls behind Methods 1, 3 and 5 for Re large.

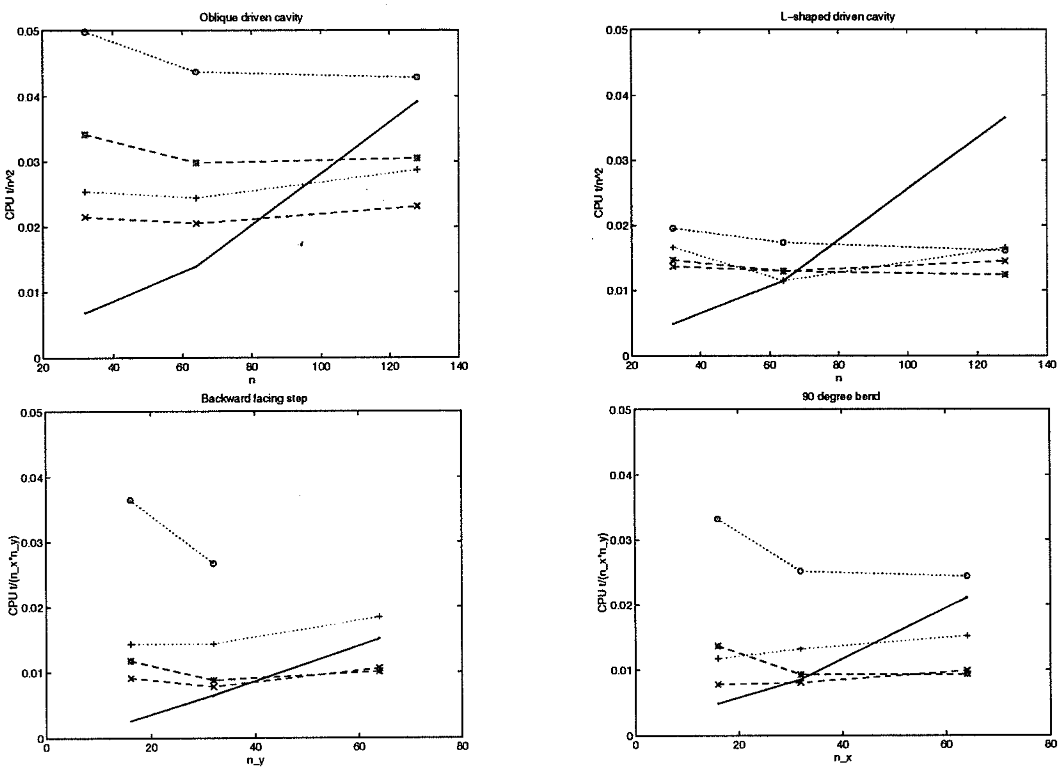


Figure 4: CPU times per grid point on the HP for the pressure equation during 40 time steps.

3.2.2 The pressure equation

The properties of the discretized pressure equation depends only on: the grid size and the shape of the space domain.

Grid size dependence

The multigrid and combined methods require the same number of iterations for increasing grid size. Again Method 1 depends on the grid size; the number of iterations grows for increasing grid size. This is illustrated by Table 1 where the results for the oblique driven cavity problem are given.

Problem dependence and comparison

The CPU time on an HP 735 per grid point for 40 time steps is shown in Figure 4. It appears that for both smoothers the combination of GCR and multigrid is more efficient then multigrid itself. Especially in the oblique driven cavity problem, Method 4 is two times as fast as Method 2. Also for the strong ILU smoother the CPU time for Method 5 is considerably less than for Method 3.

Finally, we compare Method 1 with the best multigrid method: Method 5. It appears that Method 1 is more efficient for medium grid sizes, whereas Method 5 is more efficient for large grid sizes. For the driven cavity problems the break-even point is in range [64, 128] whereas for the other problems the break-even point is in the range [32, 64]. For the pressure equation Method 1 has a superlinear convergence behaviour [12], which means the reduction of residuals is faster in later iterations than in the first ones. Since the multigrid and combined method are linear convergent, this implies that decreasing the termination criterion *tol* would benefit Method 1 and vice-versa.

3.3 Experiments on a vector machine

In this subsection we report on some experiments on a Convex C3840. First, we compare Methods 1, 3 and 5, because they are the best methods on the scalar machine and have different vectorization properties. Thereafter, Methods 3 and 5 are compared with Methods 2 and 4 to analyse the performance of methods using a weaker smoother but with greater vectorization potential and using a stronger smoother but with smaller vectorization capability.

Comparing the best methods

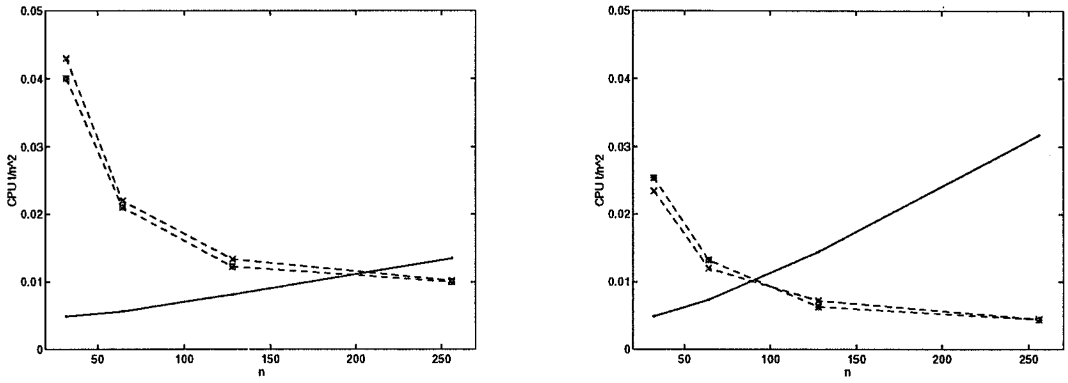


Figure 5: CPU times per grid point on the Convex during 40 time steps for the L-shaped driven cavity problem, with $Re = 1$ and $\Delta t = 0.0625$. Left: the momentum equations, right: the pressure equation.

In Figure 5 we present the CPU time per grid point against grid size for the L-shaped driven cavity problem. To show the effect of an increasing vector length, computations on a 256×256 grid are included. From this figure it appears that the convergence behaviour of the methods is comparable to that on a scalar machine: the efficiency of Method 1 deteriorates and that of Methods 3 and 5 improves with grid refinement. Due to the good vectorization properties of the Krylov methods the break-even point moves to finer grids and the GCR overhead for the combined methods becomes negligible. Finally, the curves for Methods 3 and 5 become flatter when going to finer grids, which indicates that the efficiency gain from a larger vector length is gradually exhausted.

Comparing the vectorization properties of the smoothers

It appears that the higher Mflop rate of Methods 2 and 4 does not compensate the slower rate of convergence, although on the vector machine they compete better than on the scalar machine. This is true for all test problems and is illustrated with the momentum equations of the L-shaped driven cavity problem in Table 2. Note that for a low Reynolds number Methods 2 to 5 are comparable, but for a high Reynolds number Methods 3 and 5 are superior to Methods 2 and 4. Method 2 does not work on finer grids and even fails on the 256 × 256 grid.

	Re = 1				Re = 1000			
grid size	32	64	128	256	32	64	128	256
Method 2	0.039	0.022	0.013	0.011	0.028	0.023	0.045	∞
Method 3	0.040	0.021	0.012	0.010	0.032	0.017	0.011	0.010
Method 4	0.040	0.020	0.012	0.010	0.032	0.020	0.017	0.017
Method 5	0.043	0.022	0.013	0.010	0.033	0.018	0.012	0.009

Table 2: CPU time per grid point on the Convex during 40 time steps for the momentum equations for the L-shaped driven cavity problem.

4 Conclusions

We have investigated numerically five iterative methods, namely, Method 1: GMRESR: GCR with GMRES as inner loop, Method 2: multigrid with a Jacobi line smoothing, Method 3: multigrid with an ILU smoothing, Method 4: GCR with multigrid with Jacobi line smoothing as inner loop and Method 5: GCR with multigrid with ILU smoothing as inner loop, in the context of application to the solution of the incompressible Navier-Stokes equations in general coordinates on staggered grids, using the pressure correction method in the time-dependent case.

From our numerical experiments we draw the following conclusions:

- For the solution of the momentum equations with a high Reynolds number Method 1 is the best method.
- For solving the momentum equations with a low Reynolds number Method 1 is faster for medium sized grids, whereas Method 3 is the best method for large sized grids.
- For the pressure equation Method 1 is also optimal for medium grid sizes. For large grid sizes Method 5 is the most robust and efficient method.
- The GCR outer loop of Methods 4 and 5 speeds up the rate of convergence, especially for weak smoothers (Method 4).

Finally, we remark that the break-even point, where the efficiency of the Krylov subspace method is equal to that of the multigrid method, depends on many factors. Some of them are: the domain of the test problem, the termination criterion, the Reynolds number, the computer used (scalar, vector, or parallel), etc. In Section 3 we have investigated numerically in which direction the break-even point moves depending on a change of one of these factors.

REFERENCES

[1] Eisenstat, S.C., H.C. Elman and M.H. Schultz, *Variable iterative methods for non-symmetric systems of linear equations.* SIAM J. Numer. Anal. **20**, 345–357, 1983.

[2] K.J. Morgan, J. Periaux, and F. Thomasset, editors. *Analysis of Laminar Flow over a Backward Facing Step*, Braunschweig, 1984. GAMM Workshop held at Bievres (Fr.), Vieweg.

[3] Kan, J.J.I.M. van, *A second-order accurate pressure-correction scheme for viscous incompressible flow.* SIAM J. Sci. Stat. Comput. **7**, 870–891, 1986.

[4] Oosterlee, C.W., and P. Wesseling, *A multigrid method for an invariant formulation of the incompressible Navier-Stokes equations in general co-ordinates.* Communications in Applied Numerical Methods, **8**, 721–734, 1992.

[5] Oosterlee, C.W., and P. Wesseling, *A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates*, Impact. Comp. Science Engng., **5**, 128–151, 1993 .

[6] Oosterlee, C.W., and P. Wesseling, *Multigrid schemes for time-dependent incompressible Navier-Stokes equations.* Impact. Comp. Science Engng., **5**, 153–175, 1993.

[7] Saad, Y. and M.H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems.* SIAM J. Sci. Statist. Comput. **7**, 856–869, 1986.

[8] Segal, A., P. Wesseling, J. van Kan, C.W. Oosterlee and C.G.M. Kassels, *Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates.* Int. J. Numer. Methods in Fluids **15**, 411–426, 1992.

[9] Vorst, H.A. van der and C. Vuik, *GMRESR: A family of nested GMRES methods.* Num. Lin. Alg. Appl. **1**, 369–386, 1994.

[10] Vuik, C., *Further experiences with GMRESR.* Supercomputer **55**, 13–27, 1993.

[11] Vuik, C., *Solution of the discretized incompressible Navier-Stokes equations with the GMRES method.* Int. J. Num. Methods in Fluids **16**, 507–523, 1993.

[12] Vuik, C., *Fast iterative solvers for the discretized incompressible Navier-Stokes equations.* Int. J. Num. Methods in Fluids **22**, 195–210, 1996.

[13] Wesseling, P., *An introduction to multigrid methods.* John Wiley & Sons, Chichester, 1992.

[14] Zeng, S. and P. Wesseling, *An ILU smoother for the incompressible Navier-Stokes equations in general coordinates.* Int. J. Num. Methods in Fluids **20**, 59–74, 1995.

[15] Zeng, S. and P. Wesseling, *Numerical study of a multigrid method with four smoothing methods for the incompressible Navier-Stokes equations in general coordinates.* In: N. Duane Melson, T.A. Manteuffel, S.F. McCormick (eds.): Sixth Copper Mountain Conference on Multigrid Methods. NASA Conference Pub. 3224, pp. 691-708, 1993.

[16] Zeng, S. and P. Wesseling, *Multigrid solution of the incompressible Navier-Stokes equations in general coordinates.* SIAM J. Num. Anal. **31** 1764–1784, 1994.

Page intentionally left blank

AN ALGEBRAIC MULTIGRID SOLVER FOR NAVIER-STOKES PROBLEMS IN THE DISCRETE SECOND-ORDER APPROXIMATION

R Webster
Roadside, Harpsdale, Halkirk, Caithness, KW12 6UL, Scotland, UK

ABSTRACT

An algebraic multigrid scheme is presented for solving the discrete Navier-Stokes equations to second-order accuracy using the defect-correction method. Solutions have been obtained for problems involving both structured and unstructured meshes, with the resolution and resolution grading controlled by global and local mesh refinements.

The solver is efficient and robust to the extent that no underrelaxation of variables has been required to ensure convergence, but rates of convergence can be improved with small amounts of underrelaxation of the velocity-pressure coupling. Provided that the computational mesh can resolve the flow field, convergence characteristics are almost mesh independent. Rates of convergence actually improve with refinement, asymptotically approaching mesh independent values. For extremely coarse meshes where dispersive truncation errors would be expected to prevent convergence (or even induce divergence), solutions can still be obtained by using explicit underrelaxation in the iterative cycle.

INTRODUCTION

Solution of the equations of motion for viscous fluids in the discrete approximation demands powerful computing resources. This is because the flow fields of practical interest are invariably complex and require a high degree of spatial resolution. Resolution of length scales that span many orders of magnitude may be necessary even for stable lamina flows. If Q is some measure of the linear resolving power of a discretisation (such as an appropriately scaled inverse of the nodal separation), then the number of discrete equations to be solved, N, will scale as

$$N \sim Q^d \qquad (1)$$

where d is the number of spatial dimensions. Since, moreover, the computational work will scale as N^{β}, where β depends on the solution method ($\beta > 1.0$), the required computing time, T, will scale as

$$T \sim Q^{\beta d} \tag{2}$$

Clearly T can be a very strong function of the required resolution. For example, for 3D finite-element problems that require direct solution methods (such as Gaussian elimination), the exponent can be as large as 9 (i.e., $\beta = 3$, $d = 3$). Since in fluid dynamics we are looking for orders-of-magnitude improvements in resolution it is essential to develop efficient solvers with optimum scaling ($\beta = 1.0$). It is also important that this scaling hold good for non-uniform, unstructured meshes so that the nodal economy can be maximised by matching the density of nodes to the required resolution, which may be both anisotropic and inhomogeneous.

In a previous paper [1], a new iterative solver was presented for the discrete Navier-Stokes equations in the first-order approximation which addressed these requirements. The method was based on a fully implicit Algebraic Multigrid (AMG) scheme. This paper describes changes to the scheme which can virtually eliminate the need for underrelaxation in the iterative cycle. Performance data have been obtained for a number of problems on both structured and unstructured computational meshes. Here results for the sudden-expansion test problem are presented for second-order accuracy using the defect correction method.

THE DISCRETE APPROXIMATIONS

The discrete equation sets for the flow variables are derived from a finite-volume discretisation of a finite-element mesh by enforcing the conservation of mass and momentum for an incompressible fluid. The simplest possible linear element is used : the triangle (in 2D), which is capable of giving second-order accurate equations. Control volumes are constructed around each vertex node by joining the centroid of each element to the centre of each side (Figure 1). Within any given element, just one flux value is used for the control surfaces so formed, and this is obtained by a special interpolation. The centroid provides the single interpolation point. A second discretisation within the element is used to derive the interpolation equation. Figure 2 shows three examples of the subcontrol volumes that have been used; the smallest is the one chosen for this work. The scheme is similar to those proposed by Prakash[2], Hookey[3], and Schneider and Raw[4].

If $\mathbf{v}$ represents the set of nodal velocities, $\mathbf{v}_e$ the set of interpolated velocities within elements, and $\mathbf{p}$ the set of nodal pressures, then enforcing the conservation laws for both nodal control volumes and element sub-control volumes delivers the following set of algebraic equations:

$$A(\mathbf{v}_e)\,\mathbf{v} + G\,\mathbf{p} = \mathbf{s} \tag{3}$$

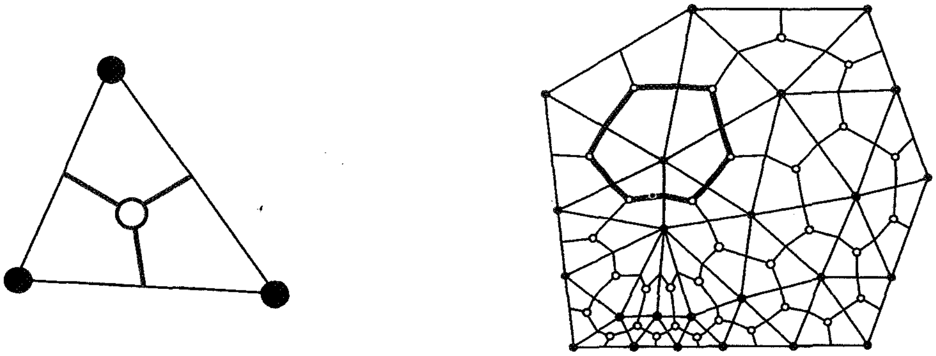


Figure 1: *Illustrating the linear triangular element, element assembly, and the construction of the control-volume tesselation; one control volume is highlighted.*

$$A_e(\mathbf{v}_e)\ \mathbf{v}_e + F(\mathbf{v}_e)\ \mathbf{v} + G_e\,\mathbf{p} = \mathbf{s}_e \qquad (4)$$

$$D\,\mathbf{v}_e = 0 \qquad (5)$$

where A and G are the nodal advection-diffusion and gradient operators respectively; A_e and F are each part of the advection-diffusion operator for elements; G_e is the element gradient operator; D is the nodal divergence operator; and s and s_e represent the momentum source/sink arrays for the nodal control volumes and for the element sub-control volumes, respectively.

The matrix A_e is diagonal, so the solution of equation (4) is trivial; that is,

$$\mathbf{v}_e = A_e^{-1}(\,\mathbf{s}_e - F\,\mathbf{v} - G_e\,\mathbf{p}) \qquad (6)$$

Direct substitution into equation (5) enables the following subset of coupled equations to be formed for the nodal variables:

$$\begin{bmatrix} A(\mathbf{v}_e) & G \\ (DA_e^{-1}F) & (DA_e^{-1}G_e) \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ (DA_e^{-1}\mathbf{s}_e) \end{bmatrix} \qquad (7)$$

The solution of equations (6) and (7) is obtained by direct iteration using a predictor-corrector strategy for $\mathbf{v}_e$ and [**v p**]; the AMG solver providing the coupled solution of equation (7) for [**v p**].

If upstream values are used in the enforcement of momentum conservation for nodal control volumes, then equation (7) will be first-order accurate. For this work, a second-order approximation is also required. The simplest possible second-order approximation was adopted using equal proportions of upstream and downstream values for the advected momentum across the control surfaces, equivalent to the central differencing of finite-difference methods.

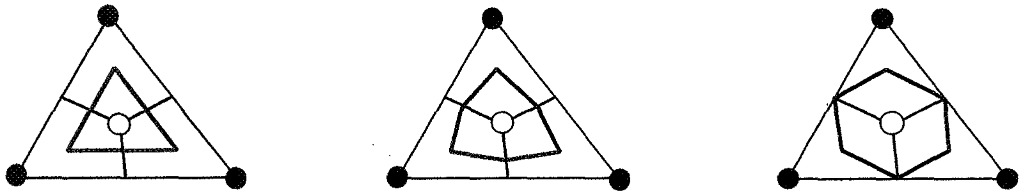


Figure 2. *Interpolation for element velocities,* $\mathbf{v}_e$ *: three subcontrol volumes that have been used for a local discrete solution of the equation of motion.*

THE ITERATIVE SOLUTION METHOD

By writing equations (6) and (7) in the more concise form as

$$\mathbf{v}_e = A_e^{-1}(s_e - H \varphi) \tag{8}$$

$$L(\mathbf{v}_e)\, \varphi = \mathbf{f} \tag{9}$$

$$\text{where}\quad L(\mathbf{v}_e) = \begin{bmatrix} A(\mathbf{v}_e) & G \\ (DA_e^{-1}F) & (DA_e^{-1}G_e) \end{bmatrix}, \quad \varphi = \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix}, \quad H = [\, F \;\; G_e \,], \quad \mathbf{f} = \begin{bmatrix} \mathbf{s} \\ (DA_e^{-1}\mathbf{s}_e) \end{bmatrix}$$

and by writing the first and second order approximations of $L(\mathbf{v}_e)$ and $\mathbf{f}$ as L_1, L_2 and $\mathbf{f}_1$, $\mathbf{f}_2$, respectively, the following iterative procedure can be constructed [5] starting with $\mathbf{v}^0{}_e = 0$ and $\varphi^0 = 0$:

$$\begin{array}{cl} \mathbf{v}_e{}^n = A_e{}^{-1} (s_e - H \varphi^n) & n > 0 \\ L_1(\mathbf{v}_e{}^n)\, \varphi^{n+1} = \mathbf{f}_1{}^n & n \leq m \\ L_1(\mathbf{v}_e{}^n)\, \varphi^{n+1} = \mathbf{f}_2{}^n + [\, L_1(\mathbf{v}_e{}^n) - L_2(\mathbf{v}_e{}^n) \,]\, \varphi^n & n > m \end{array} \tag{10}$$

where m marks a suitable point in the iteration sequence for switching on the defect correction, $[\, (L_1(\mathbf{v}_e{}^n) - \mathbf{f}_1{}^n) - (L_2(\mathbf{v}_e{}^n) - \mathbf{f}_2{}^n) \,]\varphi^n$. At convergence $\varphi^{n+1} \cong \varphi^n \cong \varphi$, and the second-order equation

$$L_2\, (\mathbf{v}_e)\, \varphi = \mathbf{f}_2{}^n \tag{11}$$

will be satisfied within the permitted tolerance. The convergence should, moreover, proceed at a rate determined more by the properties of L_1 than those of L_2 .

The equation system

$$L_1(\mathbf{v}_e{}^n)\varphi^{n+1} = \mathbf{f}^n \tag{12}$$

where $\mathbf{f}^n$ is now understood to include the defect correction if n > m, may be represented graphically as a connected nodal network with a one-to-one correspondence between variables (equations) and nodes; the connections between nodes represent the coupling between equations. For like variables, there will also be a one-to-one correspondence between connections and the edges of elements in the computational mesh. For unlike variables, connections may be regarded as displacements in an abstract dimension. To distinguish the nodal network from the computational mesh, it will be referred to as the " algebraic grid " or simply the grid.

In an iterative solution procedure based on point relaxation, each node of the grid is visited in turn and that variable is updated/corrected entirely on the basis of local information (i.e., from those neighbours to which the node has direct connections). Because of this, a single sweep through the grid system will only see changes propagating short distances (i.e., of order one nodal spacing). Long range propagation is a diffusion-like process that requires many iterative sweeps. If λ_i is a relevant propagation distance expressed in units of nodal spacing, then the number of iterations required, n, will scale as

$$n \sim \lambda_i^2 = (Q/Q_i)^2 \tag{13}$$

where Q is the maximum resolving power; Q_i is the minimum resolving power required for the resolution of λ_i. Since the computational cost of one iteration will scale as N, the total number of nodes to be visited, the required computing time will scale as

$$T \sim NQ^2 = Q^{d+2}. \tag{14}$$

Thus, from the grid system equivalent of equation (2)

$$\beta = 1 + 2/d. \tag{15}$$

Clearly, solvers based on point/local relaxation can scale poorly, with $\beta = 2$ or $\beta = 5/3$ for 2D and 3D problems, respectively. To achieve optimum $\beta = 1$ scaling it is necessary to have an efficient propagation of corrections over all length scales simultaneously. This requires multigrid methods.

AMG methods [6,7] exploit a hierarchy of reduced equation sets (coarse grids) derived from and including the base set (fine grid). Ideally, coarse grid generation proceeds recursively such that each successive grid is a consistent representation of the problem at a reduced scale of resolution, Q_i, associated length scale λ_i. Just one sweep of a relaxation procedure at this level will be sufficient to propagate changes over λ_i (i.e., $Q = Q_i$); hence, from equation (13), $n \cong 1$. With a sufficient number of grids spanning the complete range of length scales relevant to the problem, an efficient propagation over all length scales can take place simultaneously within one relaxation sweep. Thus, considering the first level of coarsening, if K is a suitably chosen restriction operator, it may be applied to the base set (12) to form the reduced system

$$L_1{}^c \varphi^c = r^c \tag{16}$$

where $L_1^c = (K L_1 K^T)$. If r^c is derived on the basis of the residual $\mathbf{r} = \mathbf{f} - L_1\varphi$:

$$r^c = Kr = K(\mathbf{f} - L_1\varphi) \qquad (17)$$

then a solution of equation (16) provides a correction φ^c that can be used to improve φ :

$$\varphi \rightarrow \varphi + K^T\varphi^c \qquad (18)$$

The procedure is as follows: restrict residual errors to the coarse grid using equation (17) ; reduce the coarse-grid (long-range) errors by applying local relaxation methods to equation set (16); prolongate the coarse-grid correction and update the fine grid solution using (18) ; and reduce the fine-grid (short-range) errors by applying local relaxation to equation set (12). Clearly equation (16) has the same form as equation (12) so the procedure can be applied recursively to generate smaller equation sets for successively coarser scale corrections. In this way a " multiscale " correction, $K^T\varphi^c$, can be assembled for updating φ.

A coarsening procedure based on that devised by Lonsdale [8] for scalar field variables has been used to generate the reduced equation sets. This consists of seeking out the equations with the strongest coupling (the largest off-diagonals in the L matrices) and joining them together by adding the corresponding matrix coefficients. Some care is required in implementing the procedure [8,1]. The elementary matrix representation of Lonsdale's restriction operator K (dimension $N_i \times N_j$, $N_i < N_j \leq N$), if required, can be formed by simply adding the appropriate rows of the $N_j \times N_j$ unit matrix. The reduction factors (N_i / N_j) may be freely chosen, though values of about 0.5 are usually used.

Since here the equation system is for coupled vector and scalar fields, the procedure is implemented in a way which preserves the block structure of the L matrix operator. Combining equations for different field variable types is thus forbidden; coarsening is only permitted in " real space ", equivalent to choosing a block-diagonal K matrix. Note that this does not prevent different coarsening for different field variables.

The process can be terminated when no further reduction in the number of equations is possible, and the matrix dimension is then equal to the number of continuum flow variables. In [1] and in this work, however, the process is actually terminated earlier at between about 30 and 60 equations.

The elementary K-matrix restriction combines equations in equal proportion. However, a better coarse grid approximation can be achieved if fine grid equations are combined in proportions that respect their relative importance at the coarser level of resolution. Therefore provision is made here for a more general, weighted restriction. For AMG solvers, this is particularly important both for uniform and non-uniform discretisations alike because, even if an initial fine grid is a regular array of identical nodes, the algebraic coarsening process is unlikely to preserve such uniformity. Thus, if R and P are the actual restriction and prolongation operators to be used, then fine grid and coarse grid weighting operators, W and W^c, are introduced such that

$$R = [W^c]^{-1} K W \tag{19}$$

subject to the scaling rule

$$R\,I\,P = I^c \tag{20}$$

where the unit operator, I , for the fine grid transforms under the action of R and P into the unit operator, I^c , for the coarse grid. Combining these equations gives

$$W^c = K\,W\,P. \tag{21}$$

For computational expediency P has been chosen to be simply K^T in this work so that the coarse grid weighting operator is simply the fine grid operator transformed using elementary restriction and prolongation.

For a finite-volume discretisation, a natural choice for W is the diagonal operator formed from the set of nodal control volumes. Equation (21) can then be simply interpreted as control-volume agglomeration and the restriction procedure R defined by equation (19) as

1. Conversion of the fine grid equations into the naturally additive net flux form (W).

2. Formation of the coarse grid equations (K,K^T).

3. A conversion of the coarse grid equations back to the normal form ($[W^c]^{-1}$).

The coarse grid approximation so produced results in a robust and an efficient solution algorithm.

Following the R-restriction of residual errors down through the grid hierarchy, with ν_1 relaxation sweeps at each level, the multiscale correction is assembled by the reverse procedure of the upward P-prolongation of solutions (possibly scaled by σ), this time applying ν_2 relaxation sweeps following each prolongation. This is the well known V-cycle schedule, $V(\nu_1,\nu_2)$. In this work, however, the full multi-grid cycle $F(\nu_1,\nu_2)$ has been adopted in which the upward leg of each cycle itself contains nested V-cycles (Figure 3). Furthermore, because the coarsest grid only contains between 30 and 60 nodes, a direct solver is used to obtain an accurate solution.

Two relaxation schemes have been adopted, both based on point Gauss-Seidel (PGS) relaxation. For the intermediate coarse grids, PGS with optimum damping is used. If $L_1{}^c = L + D + U$ is the standard splitting for Gauss-Seidel relaxation (L is the lower triangular block, U is the upper triangular block, and D the diagonal of $L_1{}^c$), then the algorithm for ν relaxation sweeps is

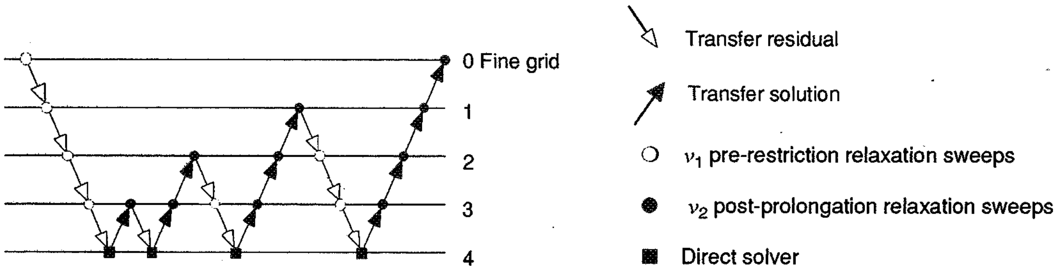

0 Fine grid
1
2
3
4
Transfer residual
Transfer solution
○ ν1 pre-restriction relaxation sweeps
● ν2 post-prolongation relaxation sweeps
■ Direct solver

Figure 3 . *F-cycle strategy for transferring residuals and corrections.*

$$\begin{aligned} \mathbf{d}^i &= (L+D)^{-1}(\,\mathbf{r}^{i-1} - U\,\mathbf{d}^{i-1}\,) \\ \mathbf{z}^i &= L_1{}^c\,\mathbf{d}^i \\ \sigma^i &= \langle\, (\mathbf{z}^i)^T, \mathbf{r}^{i-1}\,\rangle \,/\, \langle\, (\mathbf{z}^i)^T, \mathbf{z}^i\,\rangle \\ \varphi^{c(i)} &= \varphi^{c(i-1)} + \sigma^i\,\mathbf{d}^i \\ \mathbf{r}^i &= \mathbf{r}^{i-1} - \sigma^i\,\mathbf{z}^i. \end{aligned} \tag{22}$$

Before prolongation, the coarse grid corrections φ^c are also scaled by the factor

$$\sigma = \langle (L_1{}^c\varphi^c)^T, r^c\rangle \,/\, \langle (L_1{}^c\varphi^c)^T, L_1{}^c\varphi^c\rangle. \tag{24}$$

For the fine grid, an approximate 4-direction, point Gauss-Seidel algorithm for unstructured meshes is used (4-PGS). This involves some preprocessing for the formation of 4 continuous line orderings of nodes such that each node is visited once only within each line, and lines attempt, wherever possible, to pass through each node from different directions.

The residual reduction factor, or fractional error reduction for each F-cycle, μ, depends on the efficiency of the local relaxation process (smoothing) and on the quality of the coarse grid approximation [6,7,9]. Empirical μ factors are defined and results presented for several test problems.

Although L_1 does not have to be positive definite, it must have block diagonal matrices that are suitable for solution by scalar AMG methods [6] ; diagonal blocks must be at least positive semi-definite. The first-order discretisation based on the advection of upstream momentum) produces block diagonal matrices for the velocity-component equations that should satisfy that requirement. The block diagonal matrix for the pressure equations is positive semi-definite in any case.

Boundary conditions are implicitly contained in L_1. At least one pressure node is implicitly fixed in all calculations. No special measures are necessary for dealing with boundary conditions at the lower levels of the grid system. The necessary information is automatically transferred by the restriction operator.

Implicit underrelaxation of both velocity and pressure is commonly used to ensure convergence of Navier-Stokes linear solvers. For this coupled AMG linear solver, underrelaxation has not been necessary. Provided that the above described, weighting in the restriction procedure is employed, no underrelaxation has been required for any problems tackled so far. However, a small amount of underrelaxation can improve the rates of convergence for both inner and outer iterations. It can be implemented without prejudicing the long-range spatial coupling as follows. All entries in the off-diagonal blocks of L_1 are reduced by a factor ω and/or all entries in the diagonal blocks are increased by $1/\omega$, with appropriate compensations of the right hand sides of the equation sets, evaluated using previous iterates φ^n. Optimum convergence rates occur for ω values in the range $1.0 \geq \omega \geq 0.9$.

Note that it is also possible to relax the coupling between like variables by increasing just the diagonal entries of the relevant diagonal block and making the appropriate right-hand side compensations. This is not recommended. It loosens the spatial coupling that AMG is supposed to be dealing with, which results in a degradation of convergence performance (including the scaling).

PERFORMANCE

The solver has been applied to a number of well established test problems. Here flow in a channel with a sudden asymmetric expansion is presented. This problem incorporates several features of complex fluid behaviour that can present difficulties for solvers, particularly at high Reynolds numbers (e.g., singularities, recirculation, boundary layers, entering flows, outlet flows). Some of these features have been isolated for special investigation by those involved in the development of multigrid methods.

Of interest are the quality of the second-order solutions, the rates of convergence and, in particular, the mesh dependence of both of these aspects of performance. To assist in the presentation and analysis of results it will be useful to introduce mesh resolution and grading factors and to define the convergence factors.

Mesh Resolution and Grading Factors

The inverse nodal separation (linear resolution) and its variation with direction and position (grading) is used to characterize the meshes. The global extremes of the resolution and grading will be sufficient for most purposes. Thus, reference is made to the maximum linear resolving power Q, the maximum global grading factor Γ, and the maximum local grading factor γ. Q is defined as the ratio of the largest characteristic length scale divided by the closest nodal spacing. Γ is defined as the ratio of the maximum to minimum nodal separations for elements in the mesh regardless of their position. The local grading factor for any node in the mesh is the ratio of the largest to the smallest separation of the node from its immediate neighbours (i.e., for elements common to the node). Directional aspects are thus largely ignored except where reference is made to longitudinal and transverse resolution and grading factors Q_x, Γ_x, γ_{xx}, Q_y, Γ_y, and γ_{yy}, respectively. Aspect ratio γ_{xy} will also be referred to. In this case the nodal separations in any chosen element are both selected and weighted according to their degree of alignment with the relevant direction.

Convergence factors

Convergence characteristics will be quantified in terms of the convergence factor ρ^n, where

$$\rho^n = \| \delta\varphi^n \|_\infty / \| \delta\varphi^{n-1} \|_\infty \tag{24}$$

where $\delta\varphi^n$ is the multiscale correction for the iteration index n. Thus, the larger the rate of convergence, the smaller the convergence factor. The average convergence factor ρ for a sequence of N, Navier-Stokes (i.e., outer) iterations is

$$\rho = \{ \| \delta\varphi^N \|_\infty / \| \delta\varphi^0 \|_\infty \}^{1/N} = \{ \Pi_0{}^n \rho^n \}^{1/N} \tag{25}$$

The residual reduction factors, μ and μ_i, for inner iterations are defined similarly but in terms of the Euclidian norm of the residual errors, that is

$$\mu^i = \| \mathbf{r}^i \|_2 / \| \mathbf{r}^{i-1} \|_2 \tag{26}$$

where in this case $\mathbf{r}^i$ is the residual following the F-cycle, index i.

Various F-cycle schedules have been tried from F(1,0) to F(8,2). On the fine grid, $\nu_2 = 1$ actually corresponds to one application of the 4-PGS smoother.

In practice, the important convergence parameter is the fractional reduction of error per unit of computing time which may not be quite the same as the reduction of error per iteration as defined in equation (26). However, with a fixed number, ν, of F-cycles per iteration the computing time per iteration will be more or less constant; then as long as $\mu^\nu << \rho$, ρ will be equivalent to the convergence rate in time for all practical purposes. The number of F-cycles does not have to be large to satisfy this requirement. Also, there is little if anything to be gained by insisting that μ^ν be extremely small, since much of the work done will be immediately undone when the non-linear terms are updated in the outer iteration.

ASYMMETRIC SUDDEN EXPANSION TEST PROBLEM

To test the solver on a problem with inflow and outflow boundary conditions, it has been applied to the asymmetric, sudden-expansion problem. This is a high aspect ratio problem, so it offers a convenient test for the performance of the solver on meshes with highly elongated elements.

Flow enters a two-dimensional channel with a parabolic inlet velocity profile. Some distance from the inlet there is a one sided step increase in channel width to 3/2 the original. Flow separates at the re-entrant corner and a re-circulation zone is established after the step. The axial extent of the circulation is marked by the point of re-attachment, or the point at which uni-directional flow is re-established across the entire width of the channel. This depends on the Reynolds number, Re. Results have been published for Reynolds numbers up to and, in some cases, exceeding Re = 250. Re is based on step height and mean inlet velocity (note that this definition gives values 6 times smaller than those based on hydraulic diameter and maximum inlet velocity.)

A significant length of the expanded channel (exceeding 3 hydraulic diameters) needs to be modelled to ensure that the imposed outlet boundary condition does not unduly influence the behaviour upstream. Thus, the problem is bound to be one of large aspect ratio (~10) and, in view of the need for fine resolution near the point of separation, the discretisation could prove to be nodally expensive if uniform meshes are used. Thus, only non-uniform meshes have been adopted for this investigation and results for just one unstructured mesh type have been selected for presentation.

The prototype triangulation is illustrated in Figure 4. It consists of 81 proto-elements which have been assembled to give the highest resolution at the point of separation and so that the lateral resolving power Q_y is maintained moderately high up to the point of re-attachment. The actual meshes used were obtained by a q-fold nested refinement of each proto-element into as many as $q^2 = 64$ congruent triangles, giving a finest mesh of 5184 elements (2717 nodes). The mesh is anisotropic and inhomogeneous with grading factors $\gamma_{xx} = 4$, $\gamma_{yy} = 4$, $\gamma_{xy} = 5.3$, $\Gamma_x = 32$, $\Gamma_y = 4$. Dirichlet boundary conditions for velocity and free pressure boundary conditions apply on all surfaces except the outlet. The latter (continuitive and constant pressure) was placed 38 step lengths from the expansion.

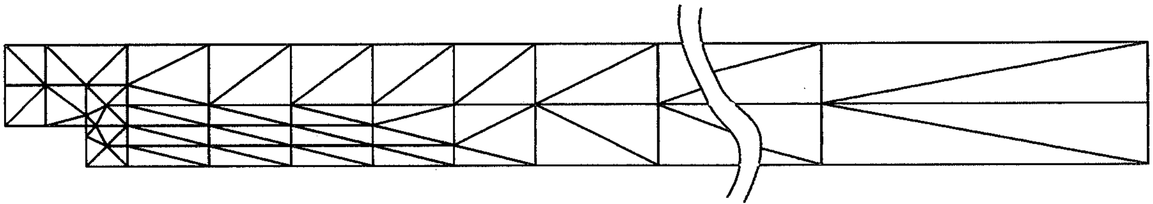


Figure 4: *Prototype triangulation for Asymmetric Sudden Expansion test problem consisting of 81 proto-elements.* $Q_x = 5\Gamma_x q$; $Q_y = 3\Gamma_y q$; *where* q = *level of nested refinement.* $\Gamma_x = 32$; $\Gamma_y = 4$; $\gamma_{xx} = 4$; $\gamma_{yy} = 4$; $\gamma_{xy} = 5.3$.

The reduction factors for this test problem were within the expected range for point Gauss-Seidel relaxation. Table 1 gives the average values for a low Reynolds number. Both definitions of Reynolds number are used (i.e., the first, Re, is based on step height and average inlet velocity, and the second, Re^h, is based on hydraulic diameter and maximum inlet velocity).

N	1236	2133	3273	4656	8151
q	3	4	5	6	8
$\mu[F(1,0)]$	.109	.159	.184	.215	.306
$\mu[F(2,1)]$	.042	.059	.091	.114	.143

Table 1: *Reduction factors for the asymmetric-sudden-expansion test problem;* Re = 16.67; Re^h = 100.

Convergence factors for the finest mesh for the same range of Reynolds numbers are presented in Table 2. This reveals slower rates of convergence; nevertheless,these rates are still better than those for segregated solution methods. In Table 3, typical values for ρ are given at four different levels of refinement at just three selected Reynolds numbers.

The convergence performance would appear to be better than that achieved by Dick and Linden [10], who obtained second-order accurate, coupled solutions to the same test problem discretised using a flux-difference splitting approach. They also used a defect-correction scheme, but their solver was based on a geometric (FAS) multigrid method. Their published result for the case corresponding here to Re = 100 was $\rho = 0.81$, which compares with $\rho = 0.68$ in Table 2. Dick and Linden also reported a deterioration in

Re	16.67	50	100	150	200
Re^h	100	300	600	900	1200
ρ	.426	.587	.684	.754	.816

Table 2: *Convergence factors for the asymmetric-sudden-expansion test problem; level of refinement* q=8; *number of unknowns* = 8151.

N	2133	3273	4656	8151
q	4	5	6	8
ρ(Re=16.7)	.464	.432	.426	.426
ρ(Re=50)	.602	.608	.587	.587
ρ(Re=150)	.911	.807	.771	.754

Table 3: *Convergence factors for the asymmetric-sudden-expansion test problem;* N = *number of unknowns*; q = *level of nested refinement.*

convergence performance with mesh refinement, which has not been observed in this work. The evidence is for constant or improving convergence rates with mesh refinement (Table 3).

Navier-Stokes Performance.

The axial extent of the recirculation eddy following the step expansion will be used as the gauge for assessing the quality of the solutions. Experimental data is available, but not for a truly parabolic inlet velocity profile. Predictions of the experiment would have to be based, therefore, on the measured profile, which is known to result in a short eddy. Since over-diffusive calculational methods would tend to underpredict the eddy length anyway, there could well be fortuitously good first-order calculations of this experiment wherever a parabolic inlet velocity profile has been mistakenly used. Here such complications are avoided by assessing the performance against other calculations of the idealised problem only. Thus the results are compared with the higher-order accurate calculations of Hutton and Smith [11] and with the first and second-order accurate calculations of Shaw [12].

For Reynolds numbers up to Re = 200, the resolution requirement should be satisfied for the mesh specified in Figure 4 (for q = 8). Results for the range Re = 16.7 to Re = 200 are given in Figure 5 as the 5 filled-circle data points. For comparison, two sets of data from Hutton and Smith are plotted, one as a continuous curve, which was obtained using a coarse mesh of 69 biquadratic rectangular elements (246 nodes), and the other as 4 open-circle data points obtained using a finer mesh of 256 quadratic triangular elements

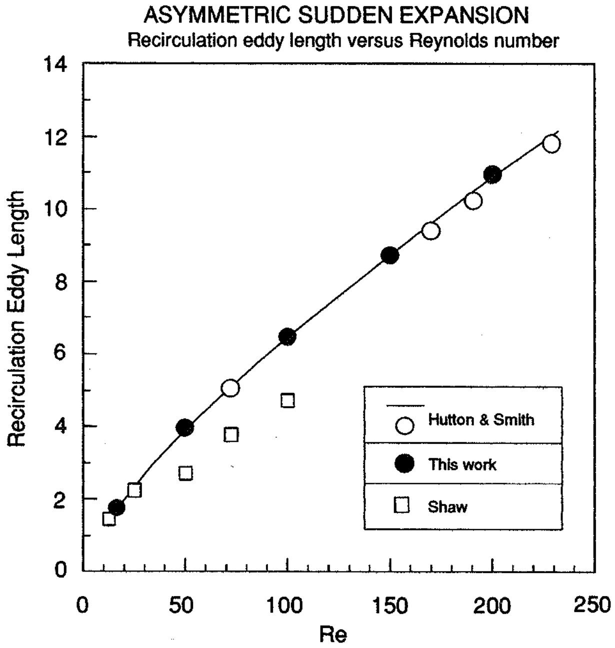


Figure 5: *Length of the recirculation eddy versus Reynolds number: a comparison with the published results of Hutton and Smith and Shaw.*

(565 nodes). The agreement is within 2% in all cases.

Five open-square data points from the calculations of Shaw, using 600 rectangular linear elements are also shown for the Reynolds number range Re = 12.5 to Re = 100. The two lower points at Re = 12.5 and Re = 25 are second order accurate and are consistent with the other data. The remaining three points were obtained using a first-order scheme for advection. They underpredict the length of the recirculation by as much as 27% at Re = 100. Shaw attributed this to the coarseness of the mesh and the false numerical diffusion associated with the first-order upwind scheme.

DISCUSSION AND GENERAL COMMENTS

The above results give a representative sample of the tests to which the solver has been applied. On the basis of all tests, the following general comments are made and the subsequent conclusions drawn.

It has not been found necessary to use any underrelaxation of variables to ensure convergence of the linear solver. The rates of reduction of the residual errors within inner iterations are typical of those to be expected for the PGS-based relaxation methods used and the simple inter-grid transfer operators being exploited. Note that, from the point of view of the coarse grid approximation, the values quoted are for the worst Navier-Stokes cases; those with low Reynolds numbers. They are nevertheless more than adequate for the problems attempted. The weak dependence of μ on mesh size is an inevitable consequence of the primitive inter-grid transfer operators used. However, it is sufficiently weak to have little if any impact on the scaling of ρ. A higher order interpolation would be required for a better coarse grid approximation, and this is unlikely to be cost effective.

Providing the computational mesh has a sufficient resolving power for the problem, rapid convergence superior to that possible with segregated solution methods is achieved. When, however, the mesh has insufficient resolution the convergence can stall ($\rho \rightarrow 1$) unless an explicit underrelaxation of velocity is exploited. This is thought to be due to the influence of the dispersive truncation error on the convergence process. For finer meshes, explicit relaxation is not required and rates of convergence improve with refinement, asymptotically approaching mesh-independent values as the resolution is increased (i.e., $\beta \rightarrow 1$ as $Q \rightarrow \infty$). No evidence has been found for $\beta > 1$ in any applications so far. If this proves to be a better performance than that achieved with other defect-correction multigrid algorithms, the accuracy of the present discretisation may be responsible.

CONCLUSIONS

An efficient and robust iterative numerical method is presented for solving the coupled equations of motion for viscous fluids in the discrete second-order approximation.

Provided that discretisation has sufficient spatial resolution for the flow field, a rapid convergence to machine accuracy is achieved that is almost mesh independent insofar as the convergence rates either improve or are maintained for increased nodal concentration.

With sufficient resolution, the method is also robust to the extent that no underrelaxation of flow variables has been required to ensure convergence. However, small amounts of underrelaxation can improve convergence rates. Converged solutions can also be obtained when the mesh resolution is insufficient to resolve the flow field, but in the more extreme cases of low resolution some explicit underrelaxation is necessary to prevent a stalling of the outer-iteration convergence.

The discretisation provides accurate solutions on relatively coarse meshes. This is probably due to the interpolation scheme used for the momentum flux within elements, which is based on a local discrete solution of the equations of motion within the element.

ACKNOWLEDGMENTS

I am grateful to Guy Robinson and Ben MacGregor for their continued interest, help, and encouragement.

REFERENCES

[1] R. Webster, " An Algebraic Multigrid Solver For Navier Stokes Problems ", *Int. j.numer. methods fluids,* 18, pp. 761-780 (1994).

[2] C. Prakash, " An Improved Control Volume Finite Element Method for Heat and Mass Transfer and for Fluid Flow Using Equal-Order,Velocity-Pressure Interpolation ", *Nvmer. Heat Transfer,* 9, pp. 253-276 (1986).

[3] N. A. Hookey, " Evaluation and Enhancements of Control Volume Finite Element Methods for Two-Dimensional Fluid Flow and Heat Transfer ", *M. Eng. thesis*, Dept. of Mech. Eng., McGill University, Montreal, Quebec, Canada (1986).

[4] G. E. Schneider and M. J. Raw, " Control Volume Finite Element Method for Heat Transfer and Fluid Flow Using Colocated Variables ", *Numer. Heat Transfer,* 11, pp. 363-390 (1987).

[5] P. W. Hemker, " Defect Correction and Higher Order Schemes for the Multigrid Solution of the Steady Euler Equations ", *Multigrid Methods II,* W. Hackbush and U. Trottenberg(eds) (*Lecture Notes in Mathematics* 1228), pp. 149-165, Springer Verlag, New York/Berlin (1986).

[6] J. Ruge and K. Stuben, " Algeraic Multigrid ", in *Multigrid Methods,* S. Cormick (ed.), Frontiers in Applied Mathematics, 5, SIAM, Philadelphia (1987).

[7] A. Brandt, " Algebraic Multigrid Theory: The Symmetric Case ", *Proceedings of the Second Copper Mountain Conference on Multigrid Methods*, Copper Mountain, Colorado (1983).

[8] R. D. Lonsdale, " An Algebraic Multigrid Solver for the Navier-Stokes Equations on Unstructured Meshes ", *Int. J. Num. Meth. Heat Fluid Flow*, 3, pp. 3-14 (1993).

[9] A. Brandt, " Rigorous Local Mode Analysis of Multigid ", *Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods*, Copper Mountain, Colorado (1989).

[10] E. Dick and J. Linden, " A Multigrid Method for Steady Incompressible Navier-Stokes Equations Based on Flux Difference Splitting ", *Int. j. numer. methods fluids,* 14, pp. 1311-1323 (1992).

[11] A. Hutton and R. Smith, " The Prediction of Laminar Flow Over a Downstream Facing Step by The Finite Element Method ", *CEGB Report RD/B/N3660*, Berkeley (1979).

[12] C. T. Shaw, " Using a Segregated Finite-Element Scheme to Solve The Incompressible Navier-Stokes Equations ", *Int. j. numer. methods fluids,* 12, pp. 81-92 (1991).

MULTIPLE COARSE GRID MULTIGRID METHODS FOR SOLVING ELLIPTIC PROBLEMS

Shengyou Xiao
Western Atlas Software
Houston,Texas

David Young
The University of Texas at Austin
Austin,Texas

April 1994

Abstract

In this paper we describe some classes of multigrid methods for solving large linear systems arising in the solution by finite difference methods of certain boundary value problems involving Poisson's equation on rectangular regions. If parallel computing systems are used, then with standard multigrid methods many of the processors will be idle when one is working at the coarsest grid levels.We describe the use of multiple coarse grid multigrid (MCGMG) methods. Here one first constructs a periodic set of equations corresponding to the given system. One then constructs a set of coarse grids such that for each grid corresponding to the grid size h there are four grids corresponding to the grid size 2*h. Multigrid operations such as restriction of residuals and interpolation of corrections are done in parallel at each grid level.For suitable choices of the multigrid operators the MCGMG method is equivalent to the parallel superconvergent multigrid (PSMG) method of Frederickson and McBryan. The convergence properties of MCGMG methods can be accurately analyzed using spectral methods.

1 Introduction

In this paper we describe some classes of multigrid methods for solving large linear systems arising from the numerical solution by finite difference methods of certain boundary value problems involving Poisson's equation

$$-u_{xx} - u_{yy} = f(x,y) \tag{1.1}$$

on rectangular domains. Here $f(x,y)$ is a given function. The solution $u(x,y)$ of (1.1.) is required to satisfy the Dirichlet condition

$$u(x,y) = g(x,y) \tag{1.2}$$

on the boundary. The standard 5-point finite difference equation is used to derive a linear system of the form

$$Au = b \tag{1.3}$$

Standard multigrid methods often exhibit excellent convergence rates on sequential computing machines. However, if parallel machines are used, many of the processors will be idle when the program is working on the coarse grid levels. Frederickson and McBryan [3] developed and analyzed a method, called the "parallel superconvergent multigrid (PSMG) method." With the PSMG method the same number of grid points are used and more of the processors are used at all grid levels. For other works dealing with the idea of using more than one coarse grid to speedup convergence cf. [2], [4], [6], [9].

In this paper we describe a class of multigrid methods which we refer to as "multiple coarse grid multigrid methods" (MCGMG methods) where, as in the case of PSMG methods, more than one coarse grid is used at each coarse grid level.

With a MCGMG method, one first constructs a periodic set of equations corresponding to the given system. One then constructs a set of coarse grids such that for each grid corresponding to the grid size h there are four grids corresponding to the grid size $2h$. The actual number of coarse grids depends on which coarsening scheme is used. There are many ways to choose the multigrid operators for a MCGMG method. For suitable choice of the operators the MCGMG method is equivalent to the PSMG method of Frederickson and McBryan. The convergence properties of MCGMG methods can be accurately analyzed using spectral methods; see, e.g., [7]. The analysis of many other iterative methods based on such a periodic set of equations can be found in, e.g., [1], [5], [8].

In Section 2, we derive Dirichlet problems and construct related discrete periodic problems corresponding to (1.1) and (1.2). In Section 3, we apply a procedure to derive a discrete periodic problem corresponding to a discrete Dirichlet problem. In Section 4, we discuss the use of MCGMG methods for solving discrete periodic problems. In Section 5, we show that a certain choice of multigrid operators can make a MCGMG method equivalent to some well known parallel multigrid methods. We also give convergence factors for the MCGMG methods and the standard multigrid methods for discrete Dirichlet problems.

It should be noted that the methods described in the paper have only been shown to apply to problems involving Poisson's equation on the rectangle with Dirichlet boundary conditions. However, it can be shown that with slight modifications, the method also applies to problems involving Neumann boundary conditions.

As pointed out by the referee, the methods used in the present paper are closely related to more general methods based on the use of symmetries; see for example [3] and the references given therein.

2 Discrete Dirichlet Problems and Discrete Periodic Problems

In this section we consider classes of discrete Dirichlet problems and discrete periodic problems in one and two dimensions. First, we consider the Dirichlet problem involving the differential equation

$$-u'' = f(x) \qquad 0 < x < 1 \tag{2.1}$$

and the boundary conditions

$$u(0) = \alpha, \quad u(1) = \beta \tag{2.2}$$

To define a discrete Dirichlet problem we choose an even positive integer N and the grid size $h = N^{-1}$ and seek a function $u(x)$ defined on the points $x = 0, h, 2h, \ldots, Nh$ such that

$$\begin{cases} 2u(x) - u(x+h) - u(x-h) = h^2 f(x) \\ \qquad\qquad x = h, 2h, \ldots, (N-1)h \\ u(0) = \alpha, \quad u(1) = \beta \end{cases} \tag{2.3}$$

For the case $N = 4$, this leads to the linear system

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \end{bmatrix} = \begin{bmatrix} h^2 f(x_1) + \alpha \\ h^2 f(x_2) \\ h^2 f(x_3) + \beta \end{bmatrix} \tag{2.4}$$

Since the matrix of the system (2.4) is nonsingular, a unique solution exists for any α, β and $f(x)$.

Let us now consider a periodic problem with period $P = 1$ based on (2.1). We require that $u(x)$ be periodic with period P and that (2.1) holds for all x. We also require that $f(x)$ be periodic with period P and that

$$\int_0^P f(x)dx = 0 \tag{2.5}$$

We now define a discrete periodic problem as follows. We require that $u(x)$ be periodic of period P on grid points $0, \pm h, \pm 2h, \ldots$, and that $u(x)$ satisfy

$$2u(x) - u(x+h) - u(x-h) = h^2 f(x), \qquad x = 0, \pm h, \pm 2h, \ldots \tag{2.6}$$

We also assume that $f(x)$ is periodic of period P and that, instead of (2.5), we have

$$\sum_0^{N-1} f_j = 0 \tag{2.7}$$

where $h = P/N$ and where $f_j = f(x_j)$, $j = 0, 1, \ldots, N-1$ and $x_j = jh$.

To actually solve the periodic problem defined by (2.6) it is sufficient to consider a finite subset of points. Thus in the case $M = 4$ we have

$$\left\{ \begin{array}{lcl} 2u_0 - u_{-1} - u_1 & = & h^2 f_0 \\ 2u_1 - u_0 - u_2 & = & h^2 f_1 \\ 2u_2 - u_1 - u_3 & = & h^2 f_2 \\ 2u_3 - u_2 - u_4 & = & h^2 f_3 \\ 2u_4 - u_3 - u_5 & = & h^2 f_4 \end{array} \right. \tag{2.8}$$

where $u_l = u(jh)$ and $f_l = f(jh)$. By periodicity we have $u_{-1} = u_3$ and $u_5 = u_1$. Thus we obtain the system

$$\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u(x_0) \\ u(x_1) \\ u(x_2) \\ u(x_3) \end{bmatrix} = \begin{bmatrix} h^2 f(x_0) \\ h^2 f(x_1) \\ h^2 f(x_2) \\ h^2 f(x_3) \end{bmatrix} \tag{2.9}$$

It can be shown that the matrix of the above system is singular and the rank is $N - 1 = 3$. Since the null space of A is spanned by the vector $(1\ 1\ 1\ 1)^T$ and since the system is consistent by (2.7), it follows that (2.9) has a solution which is unique to within an additive constant.

For general M, the eigenvalues of the operator defined by the left member of (2.6) are

$$\nu_s = 2 - 2\cos(2s\pi h); \qquad s = 0, 1, \ldots, N-1 \tag{2.10}$$

and the corresponding eigenvectors are

$$v^{(s)}(x) = e^{2\pi isx}; \qquad s = 0, 1, \ldots, N-1 \tag{2.11}$$

For the two-dimensional case we first consider the Dirichlet problem involving the Poisson equation

$$-u_{xx} - u_{yy} = f(x,y) \qquad 0 < x < 1;\ 0 < y < 1 \tag{2.12}$$

with

$$u(x,y) = g(x,y) \tag{2.13}$$

on the boundary of the square $0 \leq x \leq 1$, $0 \leq y \leq 1$. To define a discrete Dirichlet problem we choose a positive integer N and the grid size $h = N^{-1}$ and we seek a function $u(x,y)$ defined on the grid points (jh, kh), $j, k = 0, 1, \ldots, N$ such that

$$\begin{aligned}
&4u(x,y) - u(x+h,y) - u(x-h,y) \\
&\qquad -u(x,y+h) - u(x,y-h) = h^2 f(x,y) \\
&\qquad\qquad x, y = h, 2h, \ldots, (N-1)h \\
&u(x,y) = g(x,y) \\
&\qquad\qquad \begin{cases} x = 0 \text{ and } x = 1;\ y = h, 2h, \ldots, (N-1)h \\ y = 0 \text{ and } y = 1;\ x = h, 2h, \ldots, (N-1)h \end{cases}
\end{aligned} \tag{2.14}$$

Using (2.14) one obtains a linear system of the form

$$Au = b \tag{2.15}$$

where A is an $(N-1)^2$ by $(N-1)^2$ matrix. As in the one-dimensional case, the matrix A is nonsingular; hence, a unique solution to (2.15) exists.

As in the one-dimensional case we can define a discrete periodic problem with periods $P = 1$ in both the x-direction and the y-direction. We require that

$$\begin{aligned}
&4u(x,y) - u(x+h,y) - u(x-h,y) \\
&\qquad -u(x,y+h) - u(x,y-h) = h^2 f(x,y)
\end{aligned} \tag{2.16}$$

for $x, y = 0, \pm h, \pm 2h, \ldots$. Also, we assume that $f(x,y)$ is periodic with period P in x and y and that

$$\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(jh, kh) = 0 \tag{2.17}$$

It can be shown that if (2.17) holds then a solution to the discrete periodic problem defined by (2.16) exists and is unique to within an additive constant. Moreover, the eigenvalues and eigenvectors of the discrete operator defined by the left member of (2.16) are, respectively, given by

$$\nu_{s,t} = 4 - 2\cos(2\pi sh) - 2\cos(2\pi th) \tag{2.18}$$

and

$$v^{(s,t)}(x,y) = e^{2\pi isx}e^{2\pi ity}; \qquad s,t = 0,1,\ldots,N-1 \tag{2.19}$$

3 Construction of Discrete Periodic Problems

In this section we describe a procedure for constructing a discrete periodic problem corresponding to a given discrete Dirichlet problem of the type defined in Section 2.

We will illustrate the procedure for a problem in one dimension with $h = 1/4$ and $M = 4$. The procedure for the two dimensional cases is similar. From (2.4) we obtain the system

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{3.1}$$

where $f_i = f(x_i)$, $i = 1,2,3$ and

$$\begin{cases} b_1 = h^2 f_1 + \alpha \\ b_2 = h^2 f_2 \\ b_3 = h^2 f_3 + \beta \end{cases} \tag{3.2}$$

We now define $\hat{b}_i$ for $i = 0, \pm 1, \pm 2, \ldots$ as follows:

$$\begin{cases} 0 = \hat{b}_0 = \hat{b}_4 = \hat{b}_{-4} = \hat{b}_8 = \hat{b}_{-8} = \ldots \\ b_1 = \hat{b}_1 = -\hat{b}_{-1} = -\hat{b}_7 = \hat{b}_{-7} = \hat{b}_9 = -\hat{b}_{-9} = \ldots \\ b_2 = \hat{b}_2 = -\hat{b}_{-2} = -\hat{b}_6 = \hat{b}_{-6} = \hat{b}_{10} = -\hat{b}_{-10} = \ldots \\ b_3 = \hat{b}_3 = -\hat{b}_{-3} = -\hat{b}_5 = \hat{b}_{-5} = \hat{b}_{11} = -\hat{b}_{-11} = \ldots \\ \qquad\qquad \vdots \end{cases} \tag{3.3}$$

Clearly we have $\hat{b}_{j+s} = \hat{b}_j$ for $j = 0, \pm 1, \pm 2, \ldots$ and

$$\sum_{j=j^*}^{j^*+7} \hat{b}_j = 0 \tag{3.4}$$

for $j^* = 0, \pm 1, \pm 2, \ldots$.

We now consider the system

$$2w_j - w_{j+1} - w_{j-1} = \hat{b}_j, \qquad j = 0, \pm 1, \pm 2, \ldots \tag{3.5}$$

where we require that

$$w_{j+s} = w_j, \qquad j = 0, \pm 1, \pm 2, \ldots \tag{3.6}$$

It is easy to show that a necessary and sufficient condition that w is a solution of (3.5) - (3.6) is that w is a solution of the system

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} w_{-4} \\ w_{-3} \\ w_{-2} \\ w_{-1} \\ w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{b}_{-3} \\ \hat{b}_{-2} \\ \hat{b}_{-1} \\ 0 \\ \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -b_3 \\ -b_2 \\ -b_1 \\ 0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{3.7}$$

It is also easy to show that the rank of the matrix of the system (3.7) is 7 and that the null space is spanned by the vector $(1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)^T$. Therefore, because of (3.4) the system is consistent and has a solution which is unique to within an additive constant.

It should also be noted that if

$$\bar{u} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix} \tag{3.8}$$

is a solution of the original system (3.1) then $\hat{u}$ is a solution of the expanded system (3.7) where

$$\hat{u} = \begin{bmatrix} 0 \\ -\bar{u}_3 \\ -\bar{u}_2 \\ -\bar{u}_1 \\ 0 \\ \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix} \tag{3.9}$$

Let w be any solution of the expanded system (3.7). Then since (3.7) has a unique solution to within an additive constant it follows that for some constant c

$$\begin{cases} \bar{u}_1 = w_1 + c \\ \bar{u}_2 = w_2 + c \\ \bar{u}_3 = w_3 + c \end{cases} \tag{3.10}$$

If one requires that the sum of the components of w vanish, then $w = \bar{u}$ must hold, since the sum of the components of $\bar{u}$ vanishes.

We remark that the process of replacing a vector w by a vector $w' = w + c$ such that the sum of the components of w' vanishes is referred to as *purification*. Thus, if w is a vector of order N and if w' is given by

$$w'_i = w_i - \frac{1}{N}\sum_{j=1}^{N} w_j \tag{3.11}$$

for $i = 1, 2, \ldots, N$, then w' is the purified vector corresponding to w and we let

$$w' = \mathcal{P}(w) \tag{3.12}$$

4 Multiple Coarse Grid Methods

4.1 One Dimensional Case

Let $x_j = jh$ with $h = 1/N$ and

$$\Omega_h = \{x_j \mid j = 1 - N, \ldots, N\}. \tag{4.1}$$

be a grid on the interval $(-1, 1]$, where $N = 2^k$ for some positive integer k. We construct two coarse grids in such a way that all the even-numbered grid points belong to one coarse grid and all the odd-numbered grid points belong to another. Then, we have

$$\begin{aligned} \Omega_- &= \{x_j \mid x_j \in \Omega_h \text{ and } (j = \text{even})\}, & (4.2)\\ \Omega_+ &= \{x_j \mid x_j \in \Omega_h \text{ and } (j = \text{odd})\}. & (4.3) \end{aligned}$$

Figure 1 illustrates the grids on two levels, h and $2h$ for the case $N = 4$.

A two-level MCGMG algorithm for the above problem is given in Figure 2. For the following analysis, we assume that the full weighting restriction of residuals and

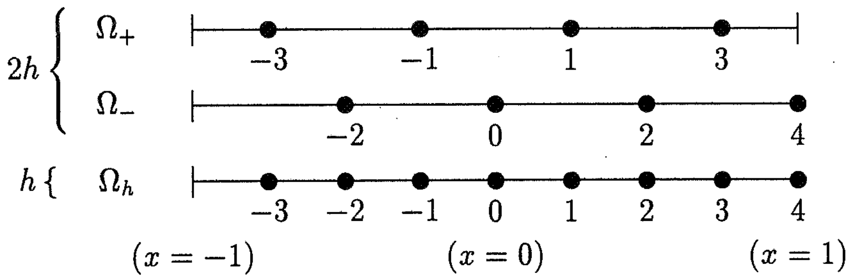


Figure 1: Two-Level Grids in 1D with $h = 1/4$

Algorithm: MCGMG2L$(A_h, u_h^{(0)}, b_h)$

1. Do m_1 pre-smoothing iterations using the smoothing iterative method (e.g., damped Jacobi method) to obtain u_h'.

2. Compute the residual $r_h = b_h - A_h u_h'$, restrict the residual onto the coarse grids and perform purification defined in (3.11) if necessary to obtain

$$r_{2h}^{(+)} = \mathcal{P}(R_h^{(+)} r_h), \quad r_{2h}^{(-)} = \mathcal{P}(R_h^{(-)} r_h)$$

   where $z_{2h}^{(+)}$ and $z_{2h}^{(-)}$ are the eigenvectors in the null spaces of $A_{2h}^{(+)}$ and $A_{2h}^{(-)}$, respectively.

3. Solve the coarse grid systems

$$A_{2h}^{(+)} \delta_{2h}^{(+)} = r_{2h}^{(+)}, \quad A_{2h}^{(-)} \delta_{2h}^{(-)} = r_{2h}^{(-)}$$

   to obtain the purified solutions $\delta_{2h}^{(+)}$ and $\delta_{2h}^{(-)}$.

4. Interpolate $\delta_{2h}^{(+)}$ and $\delta_{2h}^{(-)}$ onto the fine grid to obtain the new approximate solution

$$u_h'' = u_h' + \frac{1}{2}(P_h^{(+)} \delta_{2h}^{(+)} + P_h^{(-)} \delta_{2h}^{(-)}).$$

5. Do m_2 post-smoothing iterations using the smoothing iterative method and purify the result, if needed, to obtain $u_h^{(1)}$.

Figure 2: The 1D Two-Level MCGMG Algorithm

linear interpolation of corrections are used. The full weighting restriction is defined by

$$(R_h^{(+)} r_h)(x) = \begin{cases} \frac{1}{4}(r_h(x-h) + 2r_h(x) + r_h(x+h)) & x \in \Omega_+ \\ 0 & x \in \Omega_- \end{cases} \tag{4.4}$$

$$(R_h^{(-)} r_h)(x) = \begin{cases} 0 & x \in \Omega_+ \\ \frac{1}{4}(r_h(x-h) + 2r_h(x) + r_h(x+h)) & x \in \Omega_- \end{cases} \tag{4.5}$$

and the linear interpolation is defined by

$$(P_h^{(+)} \delta_{2h})(x) = \begin{cases} \delta_{2h}(x) & x \in \Omega_+ \\ \frac{1}{2}(\delta_{2h}(x-h) + \delta_{2h}(x+h)) & x \in \Omega_- \end{cases} \tag{4.6}$$

$$(P_h^{(-)} \delta_{2h})(x) = \begin{cases} \frac{1}{2}(\delta_{2h}(x-h) + \delta_{2h}(x+h)) & x \in \Omega_+ \\ \delta_{2h}(x) & x \in \Omega_- \end{cases} \tag{4.7}$$

The coarse grid difference operators are defined by the 3-point difference formula, e.g.,

$$\begin{aligned} (A_{2h}^{(+)} \delta_{2h}^{(+)})(x) &= (2h)^{-2}[2\delta_{2h}^{(+)}(x) - \delta_{2h}^{(+)}(x-2h) - \delta_{2h}^{(+)}(x+2h)] \\ & \quad x \in \Omega_+ \qquad (4.8) \\ (A_{2h}^{(-)} \delta_{2h}^{(-)})(x) &= (2h)^{-2}[2\delta_{2h}^{(-)}(x) - \delta_{2h}^{(-)}(x-2h) - \delta_{2h}^{(-)}(x+2h)] \\ & \quad x \in \Omega_- \qquad (4.9) \end{aligned}$$

The $2h$ coarse grids can be divided into even coarser grids in a similar way. Figure 3 illustrates all the grids on three levels, h, $2h$ and $4h$ for the case $N = 4$. Figure 4 shows the corresponding hierarchical relations among these grids.

A multilevel MCGMG algorithm is similar to the two-level version except the coarse grid problems in step 3 are solved by using algorithm MCGMG2L recursively. For a better understanding of the multilevel MCGMG algorithm, we list a three-level MCGMG algorithm in the following. For convenience of representation, we use the symbol v instead of δ to represent the solutions and b to represent the right-hand side vectors on all levels. The solutions on coarse grids should be thought of as corrections to the solution of the fine grid.

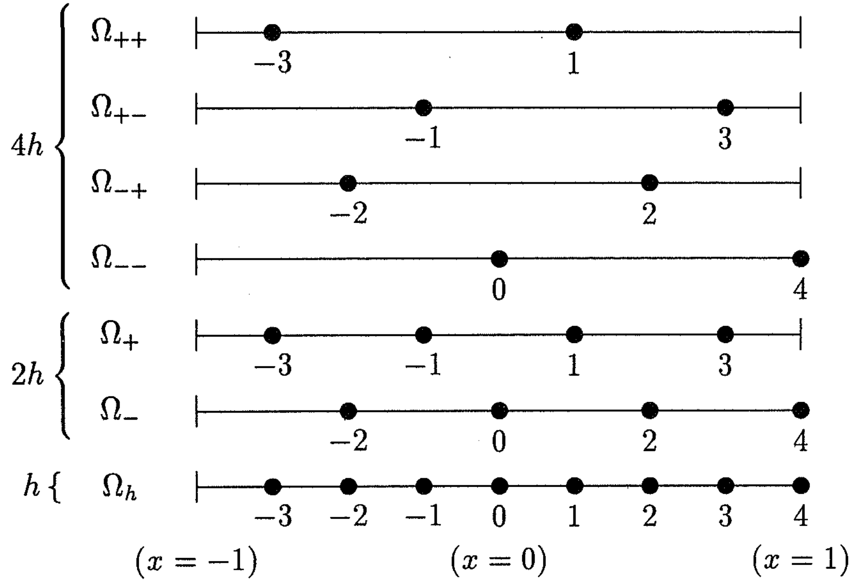


Figure 3: Coarse Grids for an Extended Fine Grid: $N = 4$

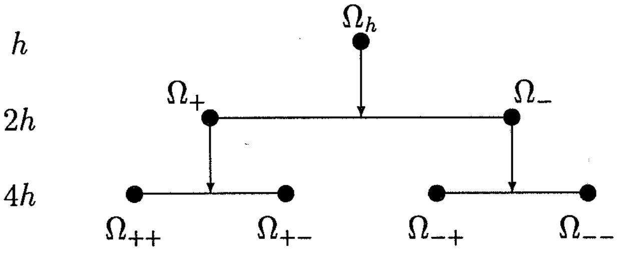


Figure 4: Hierarchical Relations Among Grids: $N = 4$

Algorithm: MCGMG1D3L$(A_h, u_h^{(0)}, b_h)$

1. Do m_1 smoothing iterations on $A_h u_h = b_h$ with initial guess v_h.

2. Compute
$$b_{2h}^{(+)} = \mathcal{P}(R_h^{(+)} r_h), \quad b_{2h}^{(-)} = \mathcal{P}(R_h^{(-)} r_h)$$

3. Do m_1 smoothing iterations on
$$A_{2h}^{(+)} u_{2h}^{(+)} = b_{2h}^{(+)}, \quad A_{2h}^{(-)} u_{2h}^{(-)} = b_{2h}^{(-)}$$
with initial guesses $v_{2h}^{(+)} = 0$ and $v_{2h}^{(-)} = 0$.

4. Compute
$$\begin{aligned} b_{4h}^{(++)} &= \mathcal{P}(R_{2h}^{(++)} r_{2h}^{(+)}), & b_{4h}^{(+-)} &= \mathcal{P}(R_{2h}^{(+-)} r_{2h}^{(+)}) \\ b_{4h}^{(-+)} &= \mathcal{P}(R_{2h}^{(-+)} r_{2h}^{(-)}), & b_{4h}^{(--)} &= \mathcal{P}(R_{2h}^{(--)} r_{2h}^{(-)}) \end{aligned}$$

5. Solve
$$\begin{aligned} A_{4h}^{(++)} u_{4h}^{(++)} &= b_{4h}^{(++)}, & A_{4h}^{(+-)} u_{4h}^{(+-)} &= b_{4h}^{(+-)} \\ A_{4h}^{(-+)} u_{4h}^{(-+)} &= b_{4h}^{(-+)}, & A_{4h}^{(--)} u_{4h}^{(--)} &= b_{4h}^{(--)} \end{aligned}$$

6. Correct
$$v_{2h}^{(+)} \leftarrow v_{2h}^{(+)} + \frac{1}{2}(P_{2h}^{(++)} v_{4h}^{(++)} + P_{2h}^{(+-)} v_{4h}^{(+-)})$$
$$v_{2h}^{(-)} \leftarrow v_{2h}^{(-)} + \frac{1}{2}(P_{2h}^{(-+)} v_{4h}^{(-+)} + P_{2h}^{(--)} v_{4h}^{(--)})$$

7. Do m_2 smoothing iterations on
$$A_{2h}^{(+)} u_{2h}^{(+)} = b_{2h}^{(+)}, \quad A_{2h}^{(-)} u_{2h}^{(-)} = b_{2h}^{(-)}$$
with initial guesses $v_{2h}^{(+)}$ and $v_{2h}^{(-)}$, respectively, and purify the results if necessary.

8. Correct
$$v_h \leftarrow v_h + \frac{1}{2}(P_h^{(+)} v_{2h}^{(+)} + P_h^{(-)} v_{2h}^{(-)})$$

9. Do m_2 smoothing iterations on $A_h u_h = b_h$ with initial guess v_h and purify the results if necessary.

Here we used the purification notation $\mathcal{P}(v,z)$ defined in (3.11) and (3.12). In the case of $N=4$, the two $2h$ coarse grid systems on the second level are given by

$$A_{2h}^{(+)}v_{2h}^{(+)} = \frac{1}{(2h)^2}\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} (v_{2h})_{-3} \\ (v_{2h})_{-1} \\ (v_{2h})_1 \\ (v_{2h})_3 \end{bmatrix}$$

$$= \begin{bmatrix} (b_{2h})_{-3} \\ (b_{2h})_{-1} \\ (b_{2h})_1 \\ (b_{2h})_3 \end{bmatrix} = \begin{bmatrix} (b_{2h}^{(+)})_{-1} \\ (b_{2h}^{(+)})_0 \\ (b_{2h}^{(+)})_1 \\ (b_{2h}^{(+)})_2 \end{bmatrix} = b_{2h}^{(+)} \tag{4.10}$$

and

$$A_{2h}^{(-)}v_{2h}^{(-)} = \frac{1}{(2h)^2}\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} (v_{2h})_{-2} \\ (v_{2h})_0 \\ (v_{2h})_2 \\ (v_{2h})_4 \end{bmatrix}$$

$$= \begin{bmatrix} (b_{2h})_{-2} \\ (b_{2h})_0 \\ (b_{2h})_2 \\ (b_{2h})_4 \end{bmatrix} = \begin{bmatrix} (b_{2h}^{(-)})_{-1} \\ (b_{2h}^{(-)})_0 \\ (b_{2h}^{(-)})_1 \\ (b_{2h}^{(-)})_2 \end{bmatrix} = b_{2h}^{(-)} \tag{4.11}$$

Here we use v_{2h} and b_{2h} to represent the fine grid vectors which consist of the coarse grid vectors $v_{2h}^{(+)}$, $v_{2h}^{(-)}$ and $b_{2h}^{(+)}$, $b_{2h}^{(-)}$, respectively.

On the third level, the four $4h$ coarse grid systems are given by

$$A_{4h}^{(++)}v_{4h}^{(++)} = \frac{1}{(4h)^2}\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} (v_{4h})_{-3} \\ (v_{4h})_1 \end{bmatrix}$$

$$= \begin{bmatrix} (b_{4h})_{-3} \\ (b_{4h})_1 \end{bmatrix} = \begin{bmatrix} (b_{4h}^{(++)})_0 \\ (b_{4h}^{(++)})_1 \end{bmatrix} = b_{4h}^{(++)}, \tag{4.12}$$

$$A_{4h}^{(+-)}v_{4h}^{(+-)} = \frac{1}{(4h)^2}\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} (v_{4h})_{-1} \\ (v_{4h})_3 \end{bmatrix}$$

$$= \begin{bmatrix} (b_{4h})_{-1} \\ (b_{4h})_3 \end{bmatrix} = \begin{bmatrix} (b_{4h}^{(+-)})_0 \\ (b_{4h}^{(+-)})_1 \end{bmatrix} = b_{4h}^{(+-)}, \tag{4.13}$$

$$\begin{aligned}A_{4h}^{(-+)}v_{4h}^{(-+)} &= \frac{1}{(4h)^2}\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} (v_{4h})_{-2} \\ (v_{4h})_2 \end{bmatrix} \\ &= \begin{bmatrix} (b_{4h})_{-2} \\ (b_{4h})_2 \end{bmatrix} = \begin{bmatrix} (b_{4h}^{(-+)})_0 \\ (b_{4h}^{(-+)})_1 \end{bmatrix} = b_{4h}^{(-+)},\end{aligned} \tag{4.14}$$

$$\begin{aligned}A_{4h}^{(--)}v_{4h}^{(--)} &= \frac{1}{(4h)^2}\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} (v_{4h})_0 \\ (v_{4h})_4 \end{bmatrix} \\ &= \begin{bmatrix} (b_{4h})_0 \\ (b_{4h})_4 \end{bmatrix} = \begin{bmatrix} (b_{4h}^{(--)})_0 \\ (b_{4h}^{(--)})_1 \end{bmatrix} = b_{4h}^{(--)}.\end{aligned} \tag{4.15}$$

Here each of the fine grid vectors v_{4h} and b_{4h} consists of four corresponding $4h$ coarse grid vectors. On the third level, the grid points on a coarse grid are not always distributed symmetrically about zero. The systems (4.12) and (4.13) may not be consistent in general. However, one can make such a problem solvable by purifying the right hand vector.

4.2 Two Dimensional Case

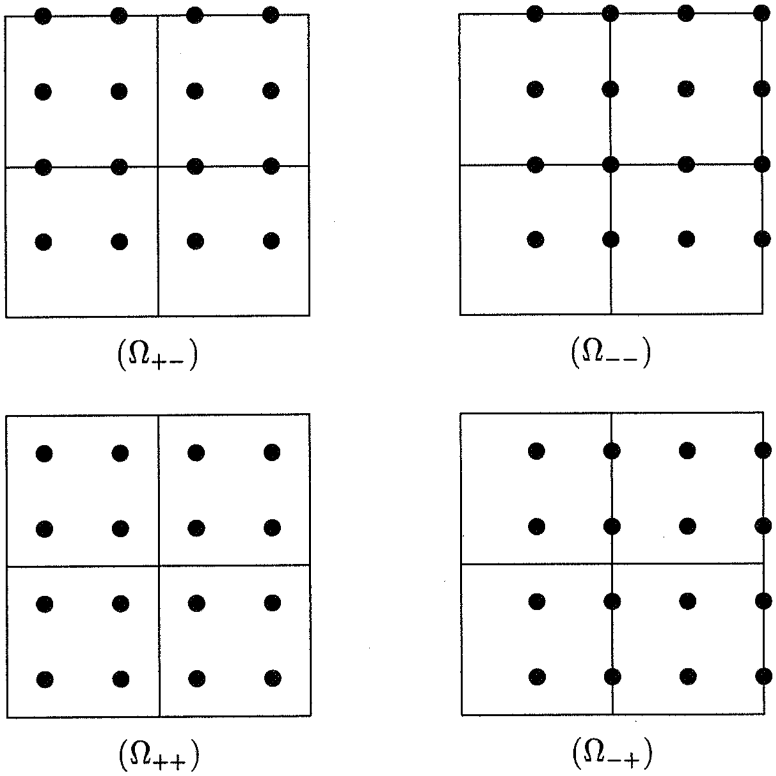
In the two dimensional region $[-1,1]^2$ we can define a grid

$$\Omega_h = \{(x_j, y_k) \mid j,k = 1-N,\ldots,N\} \tag{4.16}$$

where $x_j = jh$, $y_k = kh$ and $h = 1/N$. On this fine grid, the four coarse grids can be defined as illustrated in Figure 5 in the case of $N = 4$.

A two-level MCGMG algorithm in 2D is a straightforward extension of the corresponding two-level MCGMG algorithm in 1D defined in Figure 2. For a problem $A_h u_h = b_h$ with a given initial guess $u_h^{(0)}$, a two-level MCGMG algorithm in 2D is given in Figure 6.

As in the one dimensional case, a multilevel 2D MCGMG algorithm can be constructed by recursively applying the two-level MCGMG method to each coarse grid system until the process reaches the coarsest grid level or some preset grid level.



(Ω_{+-})

(Ω_{--})

(Ω_{++})

(Ω_{-+})

Figure 5: Coarse Grid Points for a 2D Problem with $h = 1/4$

Algorithm: MCGMG2L$(A_h, u_h^{(0)}, b_h)$

1. Do m_1 pre-smoothing iterations using the smoothing iterative method (e.g., damped Jacobi method) to obtain u_h'.

2. Compute the residual $r_h = b_h - A_h u_h'$, restrict the residual onto each of the four coarse grids and perform purification if necessary to obtain

$$r_{2h}^{(s)} = \mathcal{P}(R_h^{(s)} r_h), \qquad s = ++, -+, +-, --,$$

3. Solve the coarse grid systems

$$A_{2h}^{(s)} \tilde{\delta}_{2h}^{(s)} = r_{2h}^{(s)}, \qquad s = ++, -+, +-, --,$$

   for $\tilde{\delta_{2h}}^{(s)}$.

4. Purify $\tilde{\delta}_{2h}^{(s)}$ and interpolate the purified corrections $\delta_{2h}^{(s)}$ onto the fine grid to obtain the new approximate solution

$$\delta_{2h}^{(s)} = \mathcal{P}(\tilde{\delta}_{2h}^{(+)}, z_{2h}^{(s)}), \qquad s = ++, -+, +-, --,$$

$$u_h'' = u_h' + \frac{1}{4} \sum_s P_h^{(s)} \delta_{2h}^{(s)}.$$

5. Do m_2 post-smoothing iterations using the smoothing iterative method to obtain and return $u_h^{(1)}$.

Figure 6: The 2D Two-Level MCGMG Algorithm

5 Further Discussion

A special version of the MCGMG algorithm is determined by the selection of multigrid operations such as restriction of residuals and interpolation of corrections are done in parallel at each grid level.

For instance, if one chooses a restriction operator defined by

$$\begin{aligned}(R_h\delta_h)(x,y) \;=\; & \frac{1}{16}(\delta_h(x-h,y+h)+2\delta_h(x,y+h)+\delta_h(x+h,y+h)\\ & +2\delta_h(x-h,y)+4\delta_h(x,y)+2\delta_h(x+h,y)\\ & +\delta_h(x-h,y-h)+2\delta_h(x,y-h)+\delta_h(x+h,y-h))\\ & (x,y)\in\Omega_h \qquad (5.1)\end{aligned}$$

and an interpolation operator defined by

$$(P_h\delta_{2h})(x,y)=\delta_{2h}(x,y). \qquad (5.2)$$

then one will get a MCGMG algorithm which is equivalent to the parallel superconvergent multigrid (PSMG) method of Frederickson and McBryan [3].

One can also construct a special version of MCGMG equivalent to the frequency decomposition multigrid (FDMG) method of Hackbusch [4] by defining the coarse grid matrices

$$A_{2h}^{(s)}=R_h^{(s)}A_hP_h^{(s)} \qquad s=++,-+,+-,-- \qquad (5.3)$$

where the restriction operators $R_h^{(s)}$ are defined by

$$\begin{aligned}r_{2h}(x,y) \;=\; & (R_h^{(++)}r_h)(x,y)\\ =\; & \tfrac{1}{4}(r_h(x-h,y+h)+2r_h(x,y+h)+r_h(x+h,y+h)\\ +\; & 2r_h(x-h,y)+4r_h(x,y)+2r_h(x+h,y) \qquad (5.4)\\ +\; & r_h(x-h,y-h)+2r_h(x,y-h)+r_h(x+h,y-h))\\ & (x,y)\in\Omega_{++}.\end{aligned}$$

$$\begin{aligned}r_{2h}(x,y) \;=\; & (R_h^{(-+)}r_h)(x,y)\\ =\; & \tfrac{1}{4}(-r_h(x-h,y+h)+2r_h(x,y+h)-r_h(x+h,y+h)\\ -\; & 2r_h(x-h,y)+4r_h(x,y)-2r_h(x+h,y) \qquad (5.5)\\ -\; & r_h(x-h,y-h)+2r_h(x,y-h)-r_h(x+h,y-h))\\ & (x,y)\in\Omega_{-+}.\end{aligned}$$

$$\begin{aligned} r_{2h}(x,y) &= (R_h^{(+-)} r_h)(x,y) \\ &= \tfrac{1}{4}(-r_h(x-h,y+h) - 2r_h(x,y+h) - r_h(x+h,y+h) \\ &+ 2r_h(x-h,y) + 4r_h(x,y) + 2r_h(x+h,y) \\ &- r_h(x-h,y-h) - 2r_h(x,y-h) - r_h(x+h,y-h)) \\ &\qquad (x,y) \in \Omega_{+-}. \end{aligned} \tag{5.6}$$

$$\begin{aligned} r_{2h}(x,y) &= (R_h^{(--)} r_h)(x,y) \\ &= \tfrac{1}{4}(r_h(x-h,y+h) - 2r_h(x,y+h) + r_h(x+h,y+h) \\ &- 2r_h(x-h,y) + 4r_h(x,y) - 2r_h(x+h,y) \\ &+ r_h(x-h,y-h) - 2r_h(x,y-h) + r_h(x+h,y-h)) \\ &\qquad (x,y) \in \Omega_{--}. \end{aligned} \tag{5.7}$$

and the interpolation operators $P_h^{(s)}$ are defined by

$$\begin{aligned} \delta_h(x,y) &= (P_h^{(++)} \delta_{2h})(x,y) \\ &= \begin{cases} \delta_{2h}(x,y) & (x,y) \in \Omega_{++} \\ \frac{1}{2}(\delta_{2h}(x-h,y) + \delta_{2h}(x+h,y)) & (x,y) \in \Omega_{-+} \\ \frac{1}{2}(\delta_{2h}(x,y-h) + \delta_{2h}(x,y+h)) & (x,y) \in \Omega_{+-} \\ \frac{1}{4}(\delta_{2h}(x-h,y-h) + \delta_{2h}(x-h,y+h) & \\ +\delta_{2h}(x+h,y-h) + \delta_{2h}(x+h,y+h)) & (x,y) \in \Omega_{--}. \end{cases} \end{aligned} \tag{5.8}$$

$$\begin{aligned} \delta_h(x,y) &= (P_h^{(-+)} \delta_{2h})(x,y) \\ &= \begin{cases} \delta_{2h}(x,y) & (x,y) \in \Omega_{-+} \\ \frac{-1}{2}(\delta_{2h}(x-h,y) + \delta_{2h}(x+h,y)) & (x,y) \in \Omega_{++} \\ \frac{1}{2}(\delta_{2h}(x,y-h) + \delta_{2h}(x,y+h)) & (x,y) \in \Omega_{--} \\ \frac{-1}{4}(\delta_{2h}(x-h,y-h) + \delta_{2h}(x-h,y+h) & \\ +\delta_{2h}(x+h,y-h) + \delta_{2h}(x+h,y+h)) & (x,y) \in \Omega_{+-}. \end{cases} \end{aligned} \tag{5.9}$$

Table 1: Observed Numerical Convergence Factors

(m_1, m_2)	MCGMG	SMG
$(0,1)$	0.15	0.53
$(1,1)$	0.11	0.36
$(1,2)$	0.08	0.23

$$\delta_h(x,y) = (P_h^{(+-)}\delta_{2h})(x,y)$$
$$= \begin{cases} \delta_{2h}(x,y) & (x,y) \in \Omega_{+-} \\ \frac{1}{2}(\delta_{2h}(x-h,y) + \delta_{2h}(x+h,y)) & (x,y) \in \Omega_{--} \\ \frac{-1}{2}(\delta_{2h}(x,y-h) + \delta_{2h}(x,y+h)) & (x,y) \in \Omega_{++} \\ \frac{-1}{4}(\delta_{2h}(x-h,y-h) + \delta_{2h}(x-h,y+h) & \\ +\delta_{2h}(x+h,y-h) + \delta_{2h}(x+h,y+h)) & (x,y) \in \Omega_{-+}. \end{cases} \tag{5.10}$$

$$\delta_h(x,y) = (P_h^{(--)}\delta_{2h})(x,y)$$
$$= \begin{cases} \delta_{2h}(x,y) & (x,y) \in \Omega_{--} \\ \frac{-1}{2}(\delta_{2h}(x-h,y) + \delta_{2h}(x+h,y)) & (x,y) \in \Omega_{+-} \\ \frac{-1}{2}(\delta_{2h}(x,y-h) + \delta_{2h}(x,y+h)) & (x,y) \in \Omega_{-+} \\ \frac{1}{4}(\delta_{2h}(x-h,y-h) + \delta_{2h}(x-h,y+h) & \\ +\delta_{2h}(x+h,y-h) + \delta_{2h}(x+h,y+h)) & (x,y) \in \Omega_{++}. \end{cases} \tag{5.11}$$

corresponding to the four coarse grids Ω_{++}, Ω_{-+}, Ω_{+-} and Ω_{--}, respectively.

We used the MCGMG method to solve a test problem defined by (2.12) to (2.15) with the boundary function $g(x,y) = 1 + xy$ and grid size $h = 1/64$. The restriction operators and the interpolation operators are defined by (5.1) and (5.2) respectively. A damped Jacobi method is used for smoothing with the damping factor 0.8. For comparison, we also ran the same problem using standard multigrid method with full weighting restriction of residuals and the bilinear interpolation of corrections. Table 1 lists the observed convergence factors which are the average values of 3 cycles. The number of grid levels is 6. m_1 and m_2 are number of pre smoothing and number of post smoothing respectively. The results indicate that the observed convergence factors of a MCGMG method are much smaller than the corresponding ones of standard multigrid method.

Acknowledgments

The work was supported, in part, by the Department of Energy under Grant DE-FG03-93ER25183, by the Texas Higher Education Coordinating Board under Grant TARP-212 of the Texas Advanced Research Program, and by the National Science Foundation under Grant CCR-9504954.

REFERENCES

[1] Chan, T.F. and Elman, H.C. "Fourier Analysis of Iterative Methods for Elliptic Problems," *SIAM Review*, Vol. 31, No. 1, March 1989, pp. 20–49.

[2] Douglas, C.C. and Mandel, J. "Abstract Theory for the Domain Decomposition Method," *Computing*, Vol. 48, 1992, pp. 73-96.

[3] Frederickson, P.O. and McBryan, O. "Parallel Superconvergent Multigrid," *Multigrid Methods* (ed. S. McCormick). New York: Marcel Dekker, 1988.

[4] Hackbusch, W. "The Frequency Decomposition Multigrid Method, Part I: Application to Anisotropic Equations," *Numerische Mathematik*, Vol. 56, 1989, pp. 229–245.

[5] Kuo, C.-C. Jay and Levy, B.C. "Two-Color Fourier Analysis of the Multigrid Method with Red-Black Gauss-Seidel Smoothing," *Applied Mathematics and Computation*, Vol. 29, 1989, pp. 69–87.

[6] Mulder, Wim A. "A New Multigrid Approach to Convection Problems," *Journal of Computational Physics*, Vol. 83, 1989, pp. 303–323.

[7] Xiao, Shengyou, "Multigrid Methods with Applications to Reservoir Simulation," Report CNA-265, Center for Numerical Analysis, The University of Texas at Austin, Texas, 1994.

[8] Young, David M, Xiao, Shengyou and Baker, Karen S. "Periodically Generated Iterative Methods for Solving Elliptic Equations," *Applied Numerical Mathematics*, Vol. 19, 1995, pp. 375–387.

[9] Young, D.M. and Vona, Bi R. "Parallel Multilevel Methods," Report CNA-243, Center for Numerical Analysis, The University of Texas at Austin, March 1990.

Page intentionally left blank

NEW NONLINEAR MULTIGRID ANALYSIS*

Dexuan Xie
Courant Institute of Mathematical Sciences
New York University
251 Mercer St. New York, NY 10012

SUMMARY

The nonlinear multigrid is an efficient algorithm for solving the system of nonlinear equations arising from the numerical discretization of nonlinear elliptic boundary problems [7],[9]. In this paper, we present a new nonlinear multigrid analysis as an extension of the linear multigrid theory presented by Bramble, et al. in [5], [6], and [17]. In particular, we prove the convergence of the nonlinear *V-cycle* method for a class of mildly nonlinear second order elliptic boundary value problems which do not have full elliptic regularity.

INTRODUCTION

Multigrid methods have been used extensively to solve linear systems of equations which arise in the numerical discretization of linear partial differential equations. We call such multigrid methods "linear multigrid methods" in this paper. With the development of the linear multigrid methods, the multigrid technique also has been applied to the numerical solution of nonlinear boundary value problems. Two important algorithms have been proposed so far. One is Newton-multigrid iteration, in which a linear multigrid method is used to solve the linear system that arises from a Newton iterative method [4]. The other one is the nonlinear multigrid method, which is an extension of the linear multigrid method to the nonlinear case [9]. In literature, it is also referred to as the Full Approximation Scheme (FAS) by Brandt in [7]. The convergence of the nonlinear multigrid method was first studied by Hackbusch in [9] and later by Reusken in [11] and [12]. Hackbusch's nonlinear multigrid theory is based on his linear multigrid theory, while Reusken's analysis is based on the linear multigrid analysis in [3].

Recently, Bramble, et al. have established a new linear multigrid theory [5] [6] [17] that has generalized the work in [3] and [9] in another way. Using this new multigrid theory, they have proved the convergence of linear multigrid methods with non-nested spaces or non-inherited quadratic forms, even with weak or no regularity assumptions. The purpose of this paper is to extend this new linear multigrid theory to the nonlinear case.

In this paper, we present the framework of our new multigrid theory. In particular, we prove a basic convergence theorem for the nonlinear *V-cycle* scheme based on two abstract conditions, which are referred to as the "smoothing assumption" and the "approximation assumption".

*This work was supported in part by the National Science Foundation through award number DMS-9105437 at the University of Houston.

We then apply it to show the convergence of the nonlinear *V-cycle* method with the damped-Jacobi-Newton smoother for a class of mildly nonlinear second order elliptic boundary value problems which do not have full elliptic regularity. Moreover, our new approach makes it possible to analyze the nonlinear multigrid method in more complicated cases, such as, non-nested spaces, non-inherited quadratic forms, numerical integration, and with weak or no regularity assumptions. We have shown the convergence of the nonlinear *V-cycle* method disturbed by numerical quadratures in [14]. We intend to study other cases in subsequent work.

In comparison to the linear multigrid method, the nonlinear multigrid method has two additional parameters. In practice, their choice is an important issue. We investigate this issue numerically through a model problem in this paper. We note that this model problem, in part, aids in the understanding of the solution procedures used in the code *UHBD* [10].

The outline of the remainder of the paper is as follows. In Section 2, we introduce the basic idea of our nonlinear multigrid analysis. In Section 3, we present a general convergence theorem of the nonlinear *V-cycle* method based on two abstract assumptions, the smoothing assumption and the approximation assumption. In Section 4, we apply the theory of Section 3 to show the convergence of the nonlinear multigrid method for a class of mildly nonlinear elliptic boundary value problems. In Section 5, we present numerical experiments with the nonlinear multigrid method focusing on its two auxiliary parameters.

THE NONLINEAR MULTIGRID METHOD

We consider a nonlinear variational problem coming from a nonlinear elliptic boundary value problem with domain Ω as follows: Find $u \in H$, such that

$$a(u,v) = 0 \qquad \forall v \in H, \tag{1}$$

where $H = H(\Omega)$ is an abstract Hilbert space with inner product $(\cdot,\cdot)$, and $a(\cdot,\cdot)$ is nonlinear only with respect to the first variable.

We assume that $a(u,v)$ is H-bounded, that is, there exists a constant C, such that

$$|a(u,v)| \leq C(1+\|u\|)\|v\| \quad \forall u,v \in H,$$

where $\|u\| = \sqrt{(u,u)}$. Using the Riesz representation theorem [1], we then write (1) as

$$g(u) = 0, \tag{2}$$

where $g : H \to H$ is the nonlinear operator such that

$$a(u,v) = (g(u),v) \qquad \forall v \in H.$$

We make another assumption on g below:

A1) g is Frechet-differentiable on H, and the derivative of g at u, denoted by $Dg(u)$, is a symmetric, positive definite, bounded linear operator from H to itself.

From A1) it follows that Equation (2) has the unique solution u^* [16].

Let $\mathcal{U} \subseteq H$ be a neighborhood of u^* and $\mathcal{F}$ be the image of $\mathcal{U}$ under g. Since g satisfies the above assumptions, the implicit function theorem [1] implies that $g : \mathcal{U} \to \mathcal{F}$ is a homeomorphism. Thus, for any $f \in \mathcal{F}$, there exists unique $u \in \mathcal{U}$, such that the following equation holds:

$$g(u) = f. \tag{3}$$

Hence, we may consider equation (3) in the following.

Let u^{old} be an approximate solution of (3). The update u^{new} of u^{old} is defined by

$$u^{new} = u^{old} + q,$$

with q being a correction term satisfying the following correction equation of u^{old}:

$$g(q + u^{old}) = f. \tag{4}$$

If q is an exact solution of (4), then a direct method for solving (3) is derived. But solving (4) is as difficult as solving (3), so we often construct an approximate operator R of g^{-1} to simplify the computational work.

In the linear case, the correction equation (4) is often written as

$$g(q) = f - g(u^{old}), \tag{5}$$

and the term $f - g(u^{old})$ is often referred to as the residual of u^{old}. Clearly, if the operator R is defined by a linear iterative algorithm, then the linear iteration can be written as follows:

$$u^{new} = u^{old} + R[f - g(u^{old})]. \tag{6}$$

A key factor in the new linear multigrid theory in [5], [6] and [17] is the introduction of the operator R that characterizes the linear multigrid method, so the linear multigrid method can be expressed in form (6).

However, when g is nonlinear, the correction equation (4) cannot be written as (5). Noting the important role of the residual term in the context of the multigrid method, we introduce an "approximate" correction equation of (4) as follows:

$$g(s\hat{q} + \tilde{u}) = \tilde{f} + s[f - g(u^{old})], \tag{7}$$

where $\tilde{f} = g(\tilde{u})$, s is a given positive number and $\tilde{u}$ a given vector. Both s and $\tilde{u}$ are extra parameters, compared to the linear multigrid method, and they are chosen so that $\hat{q}$ approximates the solution q of (4) in some sense. Hence, the nonlinear multigrid method can be expressed by

$$u^{new} = u^{old} + \left[R(\tilde{f} + s[f - g(u^{old})]) - \tilde{u}\right]/s, \tag{8}$$

provided that the operator R is defined by the nonlinear multigrid iterative algorithm for solving $g(u) = f$. This is the main idea of our nonlinear multigrid analysis.

In the linear case, we can simply set $\tilde{u} = \tilde{f} = 0$ and $s = 1$. Thus, (8) reduces to (6). In this sense, the nonlinear multigrid method defined by (8) is an extension of the linear multigrid method.

To define a nonlinear multigrid operator, we need some further notation given below.

Let H be a finite element space with grid size h. Suppose that we have subspaces M_k with inner product $(\cdot,\cdot)_k$ satisfying

$$M_1 \subset M_2 \subset \cdots \subset M_l = H.$$

Set $g_l = g$, and define the nonlinear operator $g_k : M_k \to M_k$ by

$$(g_k(u), v)_k = a(u, v), \qquad \forall v \in M_k, \quad k = 1, 2, \cdots, l-1. \tag{9}$$

We define a projector $Q_k : M_{k+1} \to M_k$ by

$$(Q_k u, v)_k = (u, v)_{k+1}, \qquad \forall v \in M_k.$$

Obviously, g_k satisfies Assumption A1), so there exist $\mathcal{U}_k$ and $\mathcal{F}_k$ such that g_k is a homeomorphism between them. Hence, for $f_k \in \mathcal{F}_k$, we may consider the following equation

$$g_k(u) = f_k, \tag{10}$$

and its solution is denoted by u_k^*.

The smoothing process on M_k is denoted by the operator

$$S_k^m(\cdot; f_k) : M_k \to M_k \tag{11}$$

satisfying $u_k^* = S_k^m(u_k^*; f_k)$. We assume that S_k^m is Frechet-differentiable on M_k. Here m indicates that S_k^m may be defined by m steps of a nonlinear relaxation iteration (e.g., the damped-Jacobi-Newton or the Gauss-Seidel-Newton [13]). Without confusion, we denote $S_k^m(u; f_k)$ as $S_k^m(u)$.

Denote $\Xi_k = \{\zeta \mid \zeta = \tilde{f}_k + s_k[f_k - g_k(u_k)]$ for all $f_k \in M_k\}$. Here $\tilde{u}_k, s_k$ and u_k are fixed, and $\tilde{f}_k = g_k(\tilde{u}_k)$. We define the nonlinear multigrid operator B_k on Ξ_k inductively in the following algorithm:

Algorithm 1 *Given positive integers m_1, m_2 and p.*

0) $B_1 = g_1^{-1}$.

For each $\zeta_k \in \Xi_k$ with $k > 1$, there exists an $f_k \in M_k$ such that $\zeta_k = \tilde{f}_k + s_k[f_k - g_k(u_k)]$. We define $B_k(\zeta_k)$ in terms of B_{k-1} as follows:

1) Pre-smoothing : $v_1 = S_k^{m_1}(u_k; f_k)$.

2) Coarse grid correction: $v_2 = v_1 + \dfrac{q_p - \tilde{u}_{k-1}}{s_{k-1}}$,

where q_p is defined by (12).

$$q_i = q_{i-1} + \left[B_{k-1}(\tilde{f}_{k-1} + s_{k-1}[f_{k-1} - g_{k-1}(q_{i-1})]) - \tilde{u}_{k-1}\right]/s_{k-1}, \tag{12}$$

for $i = 1, 2, \cdots, p$. Here $q_0 = \tilde{u}_{k-1}$, and

$$f_{k-1} = \tilde{f}_{k-1} + s_{k-1}Q_{k-1}[f_k - g_k(v_1)]. \tag{13}$$

3) Post-smoothing :

$$B_k(\zeta_k) = s_k[S_k^{m_2}(v_2; f_k) - u_k] + \tilde{u}_k. \tag{14}$$

We note that Algorithm 1 using $u_k = \tilde{u}_k = 0$, $s_k = 1$, and $p = 1$ reduces to the linear multigrid algorithm described in [5], [6] and [17] provided that g is linear.

THE CONVERGENCE ANALYSIS

In our nonlinear multigrid analysis, we need a new inner product $b_k(u, v)$ defined by

$$b_k(u, v) = (Dg_k(u_k^*)u, v)_k, \quad \forall u, v \in M_k.$$

From Assumption A1) we see that $b_k(u, v)$ is symmetric, positive definite.

With this new inner product, we define an orthogonal operator $P_k : M_{k+1} \to M_k$ by

$$b_k(P_k u, v) = b_{k+1}(u, v) \qquad \forall v \in M_k.$$

From the definitions of Q_k and P_k an important equality follows:

$$Q_{k-1}Dg_k(u_k^*) = Dg_{k-1}(u_{k-1}^*)P_{k-1}, \quad k = 1, 2, \cdots, l. \tag{15}$$

Using the nonlinear multigrid operator B_k, we define the nonlinear multigrid method as follows:

$$u_k^{j+1} = \psi_k(u_k^j) \qquad j = 0, 1, 2, \cdots, \tag{16}$$

with the operator $\psi_k : M_k \to M_k$ being defined by

$$\psi_k(u_k) = u_k + \left[B_k(\tilde{f}_k + s_k[f_k - g_k(u_k)]) - \tilde{u}_k\right]/s_k. \tag{17}$$

Noting that $g_k(\tilde{u}_k) = \tilde{f}_k$ and $S_k^{m_i}(\tilde{u}_k; \tilde{f}_k) = \tilde{u}_k$ for $i = 1, 2$, we can show by induction that

$$B_k(\tilde{f}_k) = \tilde{u}_k. \tag{18}$$

Thus, the scheme (16) is consistent in the sense that u_k^* is a fixed point of the sequence $\{u_k^j\}$.

A fundamental recurrence relation with respect to the nonlinear multigrid operators B_k is given in the following theorem.

Theorem 1 *The fundamental recurrence relation for the nonlinear multigrid operators B_k, defined by Algorithm 1, is*

$$\begin{aligned} I - DB_k(\tilde{f}_k)Dg_k(u_k^*) &= DS_k^{m_2}(u_k^*)\{I - [I - (I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1}))^p] \\ &\quad Dg_{k-1}(\tilde{u}_{k-1})^{-1}Dg_{k-1}(u_{k-1}^*)P_{k-1}\}DS_k^{m_1}(u_k^*), \end{aligned} \tag{19}$$

where $k = 1, 2, \cdots, l$, and u_k^* is a solution of $g_k(u_k) = f_k$ on M_k.

Proof. Using (14), we immediately get the following equality:

$$u_k + \left[B_k(\tilde{f}_k + s_k[f_k - g_k(u_k)]) - \tilde{u}_k\right]/s_k = S_k^{m_2}(S_k^{m_1}(u_k) + \frac{q_p(u_k) - \tilde{u}_{k-1}}{s_{k-1}}), \forall u_k \in M_k. \tag{20}$$

The expression (13) of $f_{k-1}(u)$ follows

$$f_{k-1}(u_k^*) = \tilde{f}_{k-1}. \tag{21}$$

Then, by the induction and (18), we can show that

$$q_i(u_k^*) = \tilde{u}_{k-1}, \text{ for } i = 0, 1, 2, \cdots, p. \tag{22}$$

Thus, differentiating with respect to u_k at u_k^* on both sides of the equality (20), and using (22), we get

$$I - DB_k(\tilde{f}_k)Dg_k(u_k^*) = DS_k^{m_2}(u_k^*)[DS_k^{m_1}(u_k^*) + Dq_p(u_k^*)/s_{k-1}]. \tag{23}$$

Here the operations are based on the calculus in Hilbert space [1].

Using (21) and (22), we see that

$$Dq_i(u_k^*) = [I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1})]Dq_{i-1}(u_k^*) + DB_{k-1}(\tilde{f}_{k-1})Df_{k-1}(u_k^*).$$

In addition, with (13) and (15),

$$Df_{k-1}(u_k^*) = -s_{k-1}Q_{k-1}Dg_k(u_k^*)DS_k^m(u_k^*) = -s_{k-1}Dg_{k-1}(u_{k-1}^*)P_{k-1}DS_k^m(u_k^*). \tag{24}$$

Hence,

$$\begin{aligned} & Dq_p(u_k^*) = \{I + [I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1})] + \cdots \\ + \; & [I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1})]^{p-1}\}DB_{k-1}(\tilde{f}_{k-1})Df_{k-1}(u_k^*) \\ = \; & [I - (I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1}))^p]Dg_{k-1}(\tilde{u}_{k-1})^{-1}Df_{k-1}(u_k^*) \\ = \; & -s_{k-1}[I - (I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(\tilde{u}_{k-1}))^p]Dg_{k-1}(\tilde{u}_{k-1})^{-1}Dg_{k-1}(u_{k-1}^*)P_{k-1}DS_k^m(u_k^*). \end{aligned} \tag{25}$$

Therefore, the equality (19) follows by substituting (25) into (23). □

The schemes (16) with $p = 1$ and 2 are often used in practice. We refer to them as the *V-cycle* and the *W-cycle* methods, respectively. In this paper, we only consider the convergence of the nonlinear *V-cycle* method. The discussion of the other cases is similar.

Setting $p = 1$ in (19), we immediately get a fundamental recursion relation of the *V-cycle*:

$$\begin{aligned} & I - DB_k(\tilde{f}_k)Dg_k(u_k^*) \\ = \; & DS_k^{m_2}(u_k^*)[I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(u_{k-1}^*)P_{k-1}]DS_k^{m_1}(u_k^*). \end{aligned} \tag{26}$$

From the definition of $b_k(\cdot,\cdot)$, it follows that the inequality $b_k(u,u) \leq b_{k-1}(u,u)$ may not hold for some $u \in M_{k-1}$. Thus, operator $I - DB_k(\tilde{f}_k)Dg_k(u_k^*)$ may be negative with respect to the inner product $b_k(\cdot,\cdot)$. To show the convergence of the *V-cycle*, it is sufficient to prove that there exists a constant η_k in $[0,1)$, independent of h_k, such that

$$|b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u)| \leq \eta_k b_k(u,u), \quad \forall u \in M_k. \tag{27}$$

The following two basic assumptions are made to show (27):

$$\begin{aligned} |b_k((I - P_{k-1})u, u)| \; & \leq \; C_\beta^2(\frac{\|Dg_k(u_k^*)u\|_k^2}{\lambda_k})^\beta b_k(u,u)^{1-\beta}, \quad \forall u \in M_k, & (28) \\ \frac{\|Dg_k(u_k^*)u\|_k^2}{\lambda_k} \; & \leq \; C_S b_k([I - DS_k^1(u_k^*)]u, u), \qquad\qquad \forall u \in M_k, & (29) \end{aligned}$$

where λ_k is the largest eigenvalue of $Dg_k(u_k^*)$, and $0 < \beta < 1$. (28) and (29) are referred to as "the regularity and approximation assumption" and "the smoothing assumption", respectively.

The following theorem provides an estimation for a value of the parameter η_k.

Theorem 2 *Let B_k be defined by Algorithm 1 with $p = 1$ and $m_1 = m_2 = m$. Assume that*

a) Assumptions (28) and (29) hold.

b) The smoothing process S_k^m is formed by m steps of the nonlinear relaxation method S_k, such that $DS_k(u_k^*)$ is symmetric and non-negative with respect to inner product $b_k(\cdot,\cdot)$, and

$$DS_k^m(u_k^*) = [DS_k(u_k^*)]^m.$$

c) The auxiliary vector $\tilde{u}_1 = u_1^*$.

Then there exist two constants, independent of h_k,

$$\eta_{k,1} = \frac{\mathcal{M}(k)}{m^\beta + \mathcal{M}(k)} \quad \textit{and } \eta_{k,2} = 1 - \left(1 + \frac{C_\beta^2 C_S^\beta}{(2m)^\beta}\right)^k,$$

such that

$$\eta_{k,2} b_k(u,u) \le b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u) \le \eta_{k,1} b_k(u,u), \qquad \forall u \in M_k. \tag{30}$$

Furthermore, if m is sufficiently large, then the estimate (27) holds with

$$\eta_k = \max\{|\eta_{k,1}|, |\eta_{k,2}|\} < 1.$$

Here $\mathcal{M}(k)$ is a positive constant related to C_β, C_S, m, β and k. Its detail expression can be found in Theorem 1 of [5].

Proof. With $b_k(DS_k^m(u_k^*)u, v) = b_k(u, DS_k^m(u_k^*)v)$, (26) and the definition of P_{k-1}, we have

$$\begin{aligned} & b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u) = b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\ & + b_{k-1}([I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(u_{k-1}^*)]P_{k-1}DS_k^m(u_k^*)u, P_{k-1}DS_k^m(u_k^*)u). \end{aligned}$$

We now show (30) by induction on k. For $k = 1$, we have $B_1 = g_1^{-1}$ and $\tilde{u}_1 = u_1^*$. Thus,

$$|b_1([I - DB_1(\tilde{f}_1)Dg_1(u_1^*)]u, u)| = 0.$$

Suppose (30) holds for $k - 1$. We first prove the right hand side of (30). By induction,

$$\begin{aligned} & b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u) \\ \le\ & b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) + \eta_{k-1,1} b_{k-1}(P_{k-1}DS_k^m(u_k^*)u, P_{k-1}DS_k^m(u_k^*)u) \\ =\ & b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) + \eta_{k-1,1} b_k(P_{k-1}DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\ =\ & (1 - \eta_{k-1,1}) b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) + \eta_{k-1,1} b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u). \end{aligned}$$

By (28), (29) and the generalized arithmetic mean inequality,

$$\begin{aligned}
& b_k((I-P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\
\leq\ & C_\beta^2(\frac{\|Dg_k(u^*)DS_k^m(u_k^*)u\|_k^2}{\lambda_k})^\beta b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)^{1-\beta} \\
\leq\ & C_\beta^2[\beta r_k \frac{\|Dg_k(u_k^*)DS_k^m(u_k^*)u\|_k^2}{\lambda_k} + (1-\beta)r_k^{-\frac{\beta}{1-\beta}} b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)] \\
\leq\ & C_\beta^2[\beta r_k C_S b_k((I-DS_k(u_k^*))DS_k^{2m}(u_k^*)u, u) + (1-\beta)r_k^{-\frac{\beta}{1-\beta}} b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)] \\
\leq\ & C_\beta^2[\beta r_k \frac{C_S}{2m} b_k((I-DS_k^{2m}(u_k^*))u, u) + (1-\beta)r_k^{-\frac{\beta}{1-\beta}} b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)].
\end{aligned}$$

Combining the above inequalities gives

$$\begin{aligned}
& b_k([I-DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u) \\
\leq\ & [(1-\eta_{k-1,1})C_\beta^2(1-\beta)r_k^{-\frac{\beta}{1-\beta}} + \eta_{k-1,1}]b_k(DS_k^{2m}(u_k^*)u, u) \\
& +(1-\eta_{k-1,1})C_\beta^2 C_S \frac{\beta}{2m} r_k b_k([I-DS_k^{2m}(u_k^*)]u, u).
\end{aligned}$$

Now, with the same proof as that in the proof of Theorem 1 of [5], we have that

$$(1-\eta_{k-1,1})C_\beta^2(1-\beta)r_k^{-\frac{\beta}{1-\beta}} + \eta_{k-1,1} \leq \eta_{k,1}$$

and

$$(1-\eta_{k-1,1})C_\beta^2 C_S \frac{\beta}{2m} r_k \leq \eta_{k,1}.$$

This completes the proof of the right hand side of (30).

We next prove the left hand side of (30). From the spectral properties of $DS_k(u_k^*)$, it follows

$$b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \leq b_k(u,u),\ k=1,2,\cdots,l. \tag{31}$$

Combining (31) and assumptions (28) and (29) gives

$$\begin{aligned}
& -b_k((I-P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\
\leq\ & \frac{C_\beta^2 C_S^\beta}{(2m)^\beta}\left[b_k((I-DS_k^{2m}(u_k^*))u, u)\right]^\beta b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)^{1-\beta} \\
\leq\ & \frac{C_\beta^2 C_S^\beta}{(2m)^\beta}[b_k(u,u) - b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u)]^\beta b_k(u,u)^{1-\beta} \leq \frac{C_\beta^2 C_S^\beta}{(2m)^\beta} b_k(u,u),
\end{aligned}$$

where we have used the following inequality (which is similar to (3.16) in [5]):

$$b_k([I-DS_k(u_k^*))DS_k^{2m}(u_k^*)]u, u) \leq \frac{1}{2m} b_k([I-DS_k^{2m}(u_k^*)]u, u).$$

Let $\tau_k = \left(1+\frac{C_\beta^2 C_S^\beta}{(2m)^\beta}\right)^k$. By the induction assumption, we have

$$b_{k-1}([I-DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(u_{k-1}^*)]u, u) > (1-\tau_{k-1})b_{k-1}(u,u),$$

which can be written as

$$-b_{k-1}([I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(u_{k-1}^*)]u, u) < -\eta_{k-1,2}b_{k-1}(u,u).$$

Then, from the above inequalities, we obtain

$$\begin{aligned}
& -b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u) \\
= \; & -b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\
& -b_{k-1}([I - DB_{k-1}(\tilde{f}_{k-1})Dg_{k-1}(u_{k-1}^*)]P_{k-1}DS_k^m(u_k^*)u, P_{k-1}DS_k^m(u_k^*)u) \\
\leq \; & -b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) - \eta_{k-1,2}b_k(P_{k-1}DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\
= \; & -\tau_{k-1}b_k((I - P_{k-1})DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) - \eta_{k-1,2}b_k(DS_k^m(u_k^*)u, DS_k^m(u_k^*)u) \\
\leq \; & \left(\tau_{k-1}\frac{C_\beta^2 C_S^\beta}{(2m)^\beta} + \tau_{k-1} - 1\right) b_k(u,u) = (\tau_k - 1)b_k(u,u) = -\eta_{k,2}b_k(u,u).
\end{aligned}$$

The proof of the left hand side of (30) is completed. □

With Theorem 2, we now can obtain a convergence theorem of the nonlinear *V-cycle.*

Theorem 3 *Let $\{u_k^j\}$ be a sequence of iterative values of the nonlinear multigrid* V-cycle *algorithm, and let u_k^* be a solution of equation $g_k(u) = f_k$. If the assumptions in Theorem 2 hold, and m is sufficiently large, then there exists a constant σ_k with $0 < \sigma_k < 1$, independent of grid size h_k, and a neighborhood $O(u_k^*, \epsilon_k)$ of u_k^*, such that all $u_k^j \in O(u_k^*, \epsilon_k)$,*

$$\|u_k^{j+1} - u_k^*\|_{b,k} \leq \sigma_k \|u_k^j - u_k^*\|_{b,k} \quad j = 0, 1, 2, \cdots,$$

when the initial guess $u_k^0 \in O(u_k^*, \epsilon_k)$. Here $\|\cdot\|_{b,k}$, the induced norm from $b_k(\cdot,\cdot)$, is defined by $\|u\|_{b,k}^2 = b_k(u,u)$.

Proof. Clearly, from Theorem 2 it follows that

$$\|I - DB_k(\tilde{f}_k)Dg_k(u_k^*)\|_{b,k} = \sup_u \frac{|b_k([I - DB_k(\tilde{f}_k)Dg_k(u_k^*)]u, u)|}{b_k(u,u)} \leq \eta_k.$$

For a given positive number δ_k satisfying $\sigma_k = \delta_k + \eta_k < 1$, the differentiability of ψ_k at u_k^* gives that there exists a neighborhood of u_k^*, $O(u_k^*, \epsilon_k) = \{u_k : \|u_k - u_k^*\|_{b,k} \leq \epsilon_k\}$, such that

$$\|\psi_k(u_k) - \psi_k(u_k^*) - D\psi_k(u_k^*)(u_k - u_k^*)\|_{b,k} \leq \delta_k \|u_k - u_k^*\|_{b,k},$$

where $u_k \in O(u_k^*, \epsilon_k)$, ϵ_k is a positive number, and ψ_k is defined in (17). Thus

$$\begin{aligned}
& \|\psi_k(u_k) - u_k^*\|_{b,k} = \|\psi_k(u_k) - \psi_k(u_k^*)\|_{b,k} \\
& \leq \|\psi_k(u_k) - \psi_k(u_k^*) - D\psi_k(u_k^*)(u_k - u_k^*)\|_{b,k} + \|D\psi_k(u_k^*)(u_k - u_k^*)\|_{b,k} \\
& \leq (\delta_k + \|D\psi_k(u_k^*)\|_{b,k})\|u_k - u_k^*\|_{b,k} \leq \sigma_k \|u_k - u_k^*\|_{b,k}.
\end{aligned}$$

Hence, by induction, for any $u_k^0 \in O(u_k^*, \epsilon_k)$, we can easily show that $u_k^j \in O(u_k^*, \epsilon_k)$, and

$$\|u_k^{j+1} - u_k^*\|_{b,k} \leq \sigma_k \|u_k^j - u_k^*\|_{b,k} \quad j = 0, 1, 2, \cdots.$$

□

In a nonlinear multigrid algorithm, the following equations have been used on M_k for $k < l$:

$$g_k(v) = \tilde{f}_k + s_k[f_k - g_k(u_k^j)], \tag{32}$$

and

$$g_k(v) = \tilde{f}_k + s_k Q_k[f_{k+1} - g_{k+1}(v_1)], \tag{33}$$

where u_k^j is the j-th iterate of the nonlinear multigrid method, and v_1 is the iterative value after the pre-smoothing step of the nonlinear multigrid algorithm. Hence, to ensure that a nonlinear multigrid algorithm is well-defined, we should show that the solution of either (32) or (33) lies in the neighborhood $O(u_k^*, \epsilon_k)$ given in Theorem 3.

Theorem 4 *Let $O(u_k^*, \epsilon_k)$ be a neighborhood of u_k^*. Assume that*
(a) There exists a constant C such that for all $u \in M_k$ $\|Dg_k^{-1}(u)\|_{b,k} \leq C$.
(b) The auxiliary vector $\tilde{u}_k$ satisfies $\tilde{u}_k \in O(u_k^*, \epsilon_k/2)$.
(c) The auxiliary value s_k satisfies $s_k \leq \frac{\epsilon_k}{2Cr}$, when $r \neq 0$, otherwise, $s_k = 0$. Here

$$r = \max\{\|f_k - g_k(u_k^j)\|_{b,k}, \|Q_k[f_{k+1} - g_{k+1}(v_1)]\|_{b,k}\}$$

, and v_1 is the iterative value after the pre-smoothing.
Then, the solution of either (32) or (33) lies in the neighborhood $O(u_k^*, \epsilon_k)$.

Proof. We only show that the solution of (32) lies in $O(u_k^*, \epsilon_k)$. The proof for (33) is similar. Set $r_k = f_k - g_k(u_k^j)$, and $w = g_k^{-1}(\tilde{f}_k + s_k r_k)$. If $r_k = 0$, then $w = \tilde{u}_k \in O(u_k^*, \epsilon_k)$. If $r_k \neq 0$, with assumptions (a) to (c), we have

$$\begin{aligned}
\|w - u_k^*\|_{b,k} &= \|g_k^{-1}(\tilde{f}_k + s_k r_k) - u_k^*\|_{b,k} \\
&\leq \|g_k^{-1}(\tilde{f}_k + s_k r_k) - \tilde{u}_k\|_{b,k} + \|\tilde{u}_k - u_k^*\|_{b,k} \\
&= \|g_k^{-1}(\tilde{f}_k + s_k r_k) - g_k^{-1}(\tilde{f}_k)\|_{b,k} + \|\tilde{u}_k - u_k^*\|_{b,k} \\
&\leq s_k \|Dg_k^{-1}(u)\|_{b,k} \|r_k\|_{b,k} + \|\tilde{u}_k - u_k^*\|_{b,k} \\
&\leq s_k C \|r_k\|_{b,k} + \|\tilde{u}_k - u_k^*\|_{b,k} \leq \epsilon_k/2 + \epsilon_k/2 = \epsilon_k,
\end{aligned}$$

i.e. $w \in O(u_k^*, \epsilon_k)$. We complete the proof of Theorem 4. □

AN APPLICATION

In this section, as an application of the theory in Section 3, we consider the convergence of the nonlinear *V-cycle* for solving the second order elliptic, mildly nonlinear boundary value problem

$$\begin{cases} -\nabla(\alpha \nabla u) + B(x,u) = f(x), & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{34}$$

where Ω is a bounded, Lipschitz, polyhedral domain in R^d, $\alpha \in W^{1,\infty}(\Omega)$, $\alpha \geq C_\alpha > 0$ a.e. on Ω, and $f \in L^2(\Omega)$.

Let D_2B denote the derivative of $B(\cdot,\cdot)$ with respect to the second variable. We make the following assumptions on D_2B in this section.

A2) $D_2B(x,u)$ is continuous in $\bar{\Omega} \times R$, and there exist constants C_1 and C_2 such that

$$0 < C_2 \leq D_2B(x,u) \leq C_1.$$

A3) $D_2B(x,u)$ satisfies a Lipschitz condition: there exists a constant L, independent of u and v, such that

$$|D_2B(x,u) - D_2B(x,v)| \leq L|u-v|, \tag{35}$$

for all (x,u), (x,v) on a subset of $\bar{\Omega} \times R$.

Let $H = H_0^1(\Omega)$ be the Sobolev space [2]. The weak form of (34) is thus: Find $u \in H$, such that

$$a(u,v) = (f,v)_{L^2}, \qquad \forall v \in H \tag{36}$$

where

$$a(u,v) = \int_\Omega [\alpha \bigtriangledown u \bigtriangledown v + B(x,u)v]dx, \text{ and } (f,v)_{L^2} = \int_\Omega f(x)v(x)dx. \tag{37}$$

Let M_k be a set of piecewise linear functions with respect to a quasi-uniform triangulation $\mathcal{F}_k$ on Ω of size h_k in the usual sense [8]. We assume that there is a constant c , independent of k, such that $h_{k-1} \leq ch_k$, and these triangulations should be nested in the sense that any triangle in $\mathcal{F}_{k-1}$ can be written as a union of triangles of $\mathcal{F}_k$.

The finite element discretization for (36) on each M_k is as follows: *Find $u_k \in M_k$ such that*

$$a(u_k,v) = (f,v)_{L^2}, \qquad \forall v \in M_k \ , \tag{38}$$

where $k = 1,2,\cdots,l$.

Based on Theorem 39.12 in [16], we assume that

A4) Equations (36) and (38) have unique solutions u^* and $\hat{u}_k$, respectively. For $u^* \in H^{1+\beta}(\Omega)$ with $\beta \in (0,1]$, there exists a constant c, independent of h_k, such that

$$||u^* - \hat{u}_k||_1 \leq ch_k^\beta, \tag{39}$$

where $k = 1,2,\cdots,l$, and $||\cdot||_1$ is the usual norm in Sobolev space H^1 [2].

We solve equation (38) by the nonlinear multigrid *V-cycle* scheme with the smoother S_k^m defined by m steps of the damped-Jacobi-Newton iteration. To prove its convergence, using Theorem 3, we only need to verify Assumptions (29) and (28).

We first prove Assumption (29) for the smoother S_k^m below.

Let $\{\varphi_i\}_{i=1}^{n_k}$ be a natural nodal basis for M_k, where $n_k = dim M_k$. Apparently, we may consider the following equation on M_k: For $f_k \in M_k$, find $u_k \in M_k$ such that

$$(g_k(u_k),\varphi_\nu)_k = (f_k,\varphi_\nu)_k, \quad \nu = 1,2,\cdots,n_k,$$

with g_k being defined by

$$(g_k(u_k),v)_k = a(u_k,v) - (f,v)_{L^2}, \qquad \forall v \in M_k. \tag{40}$$

Let u_k^j be the j-th iterate of the damped-Jacobi-Newton iteration using a damping parameter θ, expressed as follows:

$$u_k^{j+1} = u_k^j + R_k(u_k^j)[f_k - g_k(u_k^j)],$$

where the linear operator $R_k(u) : M_k \to M_k$ is defined by

$$R_k(u)v = \theta \sum_{i=1}^{n_k} \left(\frac{\partial g_k(u)}{\partial u_{k,i}} \varphi_i, \varphi_i \right)_k^{-1} (v, \varphi_i)_k \varphi_i \quad \forall v \in M_k.$$

Since $S_k^1(u) = u + R_k(u)(f_k - g_k(u))$, and

$$DS_k^1(u_k^*) = I - R_k(u_k^*)Dg_k(u_k^*), \tag{41}$$

we have

$$DS_k^m(u_k^*) = [I - R_k(u_k^*)Dg_k(u_k^*)]^m = [DS_k^1(u_k^*)]^m.$$

Clearly, $DS_k^1(u_k^*)$ is symmetric, so Assumption b) of Theorem 2 holds. From (41) we see that the Jacobi-Newton iteration has a similar form as the damped-Jacobi method in [17]. Therefore, using the same argument as in [17], we can show that Assumption (29) is satisfied by the damped-Jacobi-Newton iteration.

We next verify Assumption (28). Let g be defined by

$$(g(u), v) = a(u, v) - (f, v)_{L^2}, \qquad \forall v \in H. \tag{42}$$

It is easy to show that $Dg(w)$, defined by

$$(Dg(w)u, v) = \int_\Omega [\alpha \bigtriangledown u \bigtriangledown v + D_2B(x, w)uv]dx, \quad \forall v \in H,$$

is symmetric, positive definite on H.

Hence, from (40) it follows that $Dg_k(w)$ is a symmetric, positive definite operator on M_k. Thus, the bilinear form on $M_k \times M_k$

$$b_k(u, v) \equiv (Dg_k(w)u, v)_k, \;\; \forall u, v \in M_k, \tag{43}$$

is symmetric, positive definite.

For simplicity, we let $A_k \equiv Dg_k(u_k^*)$, and define a family of norms as follows:

$$|||v|||_{r,k}^2 = (A_k^r v, v)_k, \qquad \forall v \in M_k,$$

where r is a positive number. In addition, we note that $|||v|||_{0,k}$ is equivalent to $||v||_{L^2}$ and $|||v|||_{1,k} = ||v||_{b,k}$.

We now can show that Assumption (28) holds in the following theorem. The proof of this theorem can be found in [15].

Theorem 5 *Let M_k be the space of continuous piecewise linear functions with respect to a quasi-uniform triangulation, and let u_k^* be the solution of equation $g_k(u) = f_k$ in M_k. Assume that (A1) to (A4) hold, and that the solution U of the variational problem*

$$b_k(U, v) = (F, v)_{L^2}, \qquad \forall v \in H \tag{44}$$

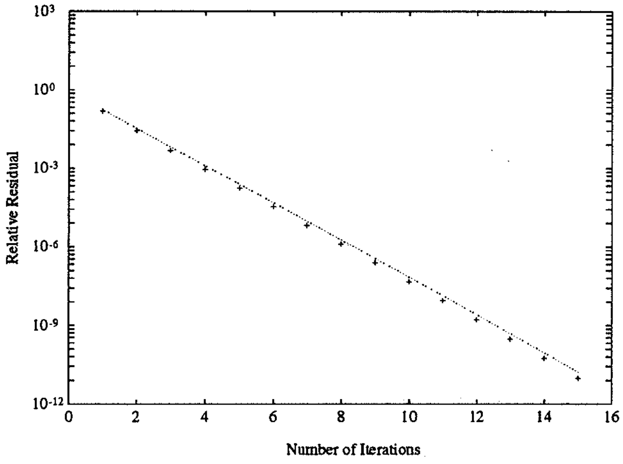


Figure 1: A comparison of a nonlinear *V-cycle* and a linear *V-cycle*. Here $\cdots$: the linear *V-cycle* method for solving (46) with $b = 0$, $+++$: the nonlinear *V-cycle* method for solving (46) with $a = b = 1$, and $h = \frac{1}{128}$.

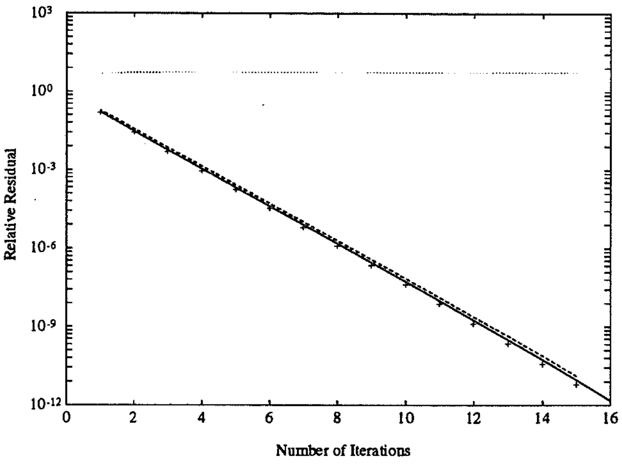


Figure 2: Dependency of the convergence rate of the nonlinear *V-cycle* on the auxiliary vector. Here $+++ : \tilde{u}_k = 0$, —: $\tilde{u}_k = Q_k u_{k+1}^{j,s}$, $--$: $\tilde{u}_k = S_k^{200}(0)$, $\cdots$: $\tilde{u}_k = 0.5$, $h = \frac{1}{128}$, and $a = b = 1$ in (46).

is in $H^{1+\beta}(\Omega)$ *for some* $\beta \in (0,1]$, *and satisfies*

$$\|U\|_{H^{1+\beta}} \leq C\|F\|_{H^{\beta-1}} \tag{45}$$

for some positive constant C, *independent of* F. *Then, there exists a constant* C *such that*

$$|b_k((I - P_{k-1})u, u)| \leq C\left(\frac{\|Dg_k(u_k^*)u\|_k^2}{\lambda_k}\right)^{\frac{\beta}{2}} b_k(u,u)^{1-\frac{\beta}{2}}, \quad \forall u \in M_k,$$

where λ_k *is the largest eigenvalue of* $Dg_k(u_k^*)$.

NUMERICAL EXPERIMENTS

In this section, we present numerical experiments with the nonlinear multigrid method for solving the following model problem [10]:

$$\begin{cases} -(u_{xx} + u_{yy}) + b\sinh(au) & = f \text{ in } \Omega = (0,1) \times (0,1), \\ u & = 0 \text{ on } \partial\Omega, \end{cases} \tag{46}$$

where a and b are positive numbers. The right hand side term f of (46) is chosen such that $u = \sin \pi x \sin \pi y$ is the solution.

The discretization equation of (46) is defined by the five-point stencil with $h_k = 1/2^k$ $(1 \leq k \leq l)$. The smoothing process S_k^m consists of m steps of the Gauss-Seidel-Newton iteration.

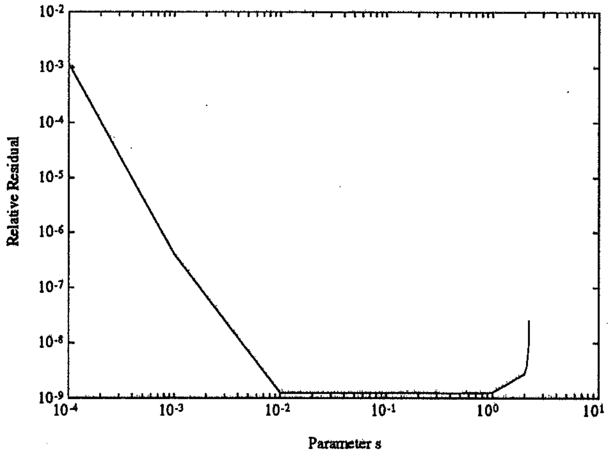


Figure 3: The relation of the relative residual of the nonlinear *V-cycle* with parameter s_k at the $12th$ *V-cycle* iteration. This figure shows that as s_k is around 1, the nonlinear *V-cycle* has an almost same convergent rate. Here $h = \frac{1}{64}$, and $a = b = 1$ in (46).

We set $m_1 = m_2 = m$ for all grid levels and the coarsest grid size $h_1 = \frac{1}{2}$ for all of our numerical examples. Besides, the full-weighting restriction operator Q_k, [9], was used, and only one step of the Gauss-Seidel-Newton iteration was applied to get the solution of the equation on the coarsest grid M_1. The initial guess $u_h^0 = 0$ and the relative residual stopping criterion were taken for all the numerical experiments, which were implemented on a $KSR1$ supercomputer with single precision, which is equal to the regular double precision.

We compared the performance of the nonlinear *V-cycle* with the linear *V-cycle* method. The linear *V-cycle* case was obtained from the nonlinear *V-cycle* program by setting $b = 0$ in (46). Thus, a Poisson equation was solved by the linear *V-cycle* method. From Figure 1 we see that the nonlinear multigrid method is as efficient as the linear multigrid method. We checked the dependency of the convergence rate of the nonlinear multigrid method on its two parameters $\tilde{u}$ and s_k. We used three different values of $\tilde{u}_k$ in the experiments.

1) $\tilde{u}_k = 0$ on all grid levels;

2) $\tilde{u}_k = S_k^m(0)$, i.e. $\tilde{u}_k$ is defined by m steps of the Gauss-Seidel-Newton iteration with zero initial guess. Clearly, by increasing m, we can make $\tilde{u}_k$ approach to the exact solution $g_k(u) = f_k$ as closely as desired.

3) $\tilde{u}_k = Q_k u_{k+1}^{j,s}$, where $u_{k+1}^{j,s}$ denotes the iterative value after the pre-smoothing step of the *V-cycle*. We call this type of $\tilde{u}_k$ Brandt's choice because it was first used by Brandt in [7]. Figure 2 shows that if $\tilde{u}_k$ is properly close to the solution of $g_k = f_k$, the convergence rate of the *V-cycle* will be almost the same. Otherwise, the nonlinear *V-cycle* may be divergent. For example, from this figure we see that the *V-cycle* with $\tilde{u}_k = 0.5$ was divergent.

For fixed $\tilde{u}_k = 0$, we also made experiments with different values of s_k. Figure 3 shows that it is satisfactory to let s_k be around 1.

Finally, we checked the influence of the a and b in (46) on the convergence of the nonlinear *V-cycle* method. The numerical results are reported in Tables 1 to 3. Here we used four different $\tilde{u}_k$, $h = \frac{1}{64}$ and $m_1 = m_2 = 1$ for all of these numerical experiments. We also used $a = 1.0$,

$b = 1.0$ and $a = 3.0$ in Table 1, Table 2 and Table 3, respectively. The notation — in the tables means that the *V-cycle* is divergent. From these tables we see that: 1) When $0 \le a < 3$ and $0 \le b < 10$, $\tilde{u}_k = 0$ is the simplest choice; 2) Brandt's choice worked for $0 \le a \le 6$ and $0 \le b \le 100$; and 3) the nonlinear *V-cycle* with $\tilde{u}_k = S_k^m(0)$ using large m can lead to convergence for a pair of a and b for which the nonlinear *V-cycle* with Brandt's choice is divergent.

Table 1: *The performance of the nonlinear* V-cycle *as the b in (46) becomes larger.*

	The Total number of Iterations			
b	$\tilde{u}_k = 0$	$\tilde{u}_k = Q_k u_{k+1}^{j,s}$	$\tilde{u}_k = S_k^1(0)$	$\tilde{u}_k = S_k^{10}(0)$
10	13	14	13	14
30	40	13	14	13
100	—	12	35	13

Table 2: *The performance of the nonlinear* V-cycle *as the a in (46) becomes larger.*

	The number of Iterations			
a	$\tilde{u}_k = 0$	$\tilde{u}_k = Q_k u_{k+1}^{j,s}$	$\tilde{u}_k = S_k^1(0)$	$\tilde{u}_k = S_k^{10}(0)$
0.001	14	14	14	14
2.0	13	14	14	14
3.0	32	14	14	15
6.0	—	12	—	30
7.0	—	—	—	20

Table 3: *The performance of the nonlinear* V-cycle *for solving (46) with large a and b.*

	The number of Iterations			
b	$\tilde{u}_k = 0$	$\tilde{u}_k = Q_k u_{k+1}^{j,s}$	$\tilde{u}_k = S_k^1(0)$	$\tilde{u}_k = S_k^{10}(0)$
0.01	14	14	14	14
1.0	32	14	14	15
20.0	—	12	—	16

ACKNOWLEDGMENTS

The author would like to thank his advisor Professor L. Ridgway Scott for valuable discussions and his continuous support. He also wants to thank Professor Olof Widlund and Professor Shuzi Zhou for their helpful comments and valuable suggestions.

REFERENCES

[1] RALPH. ABRAHAM, *Manifolds, Tensor Analysis, and Applications.* Addison-Wesley Publishing Company, Inc. , 1983.

[2] R. A. ADAMS: *Sobolev Spaces,* Academic Press, New York, 1975.

[3] R. E. BANK, AND C. C. DOUGLAS: *Sharp estimates for multigrid rates of convergence with general smoothing and acceleration.* SIAM J. Numer. Anal. 22 (1985), 617-633.

[4] R. E. BANK, D. J. ROSE,: *Analysis of a Multilevel Iterative Method for Nonlinear Finite Element Equations,* Math. Comp. 39 (1982), 453-465.

[5] J. H. BRAMBLE AND J. E. PASCIAK: *New Convergence Estimates for Multigrid Algorithms.* Math. Comp. 49 (1987), No.180.

[6] J. H. BRAMBLE AND J. E. PASCIAK: *New Estimates for Multilevel Algorithms Including the V-cycle,* Math. Comp. 60 (1993), 447-471.

[7] A. BRANDT: *Guide in Multigrid Development.* Lect. Notes in Math., 960 (1982), Springer.

[8] P. C. CIARLET: *The Finite Element Methods for Elliptic Problems,* North-Holland, Amsterdam, New York, Oxford, 1978.

[9] W. HACKBUSCH: *Multigrid Methods and Applications.* Springer Heidelberg, 1985.

[10] M. HOLST AND F. SAIED: *Multigrid solution of the Poisson-Boltzmann equation,* Journal of Computational Chemistry, 14 (1993), 105-113.

[11] ARNOLD REUSKEN: *Convergence of the Multigrid Full Approximation Scheme for a Class of Elliptic Mildly Nonlinear Boundary Value Problems,* Numer. Math. 52 (1988), 251-277.

[12] ARNOLD REUSKEN: *Convergence of the Multilevel Full Approximation Scheme Including the* V-cycle, Numer. Math. 53 (1988), 663-686.

[13] ORTEGA J. M. AND W. C. RHEINHOLDT: *Iterative Solution of Nonlinear Equations in Several Variables,* Academic press, New York, 1970.

[14] L. R. SCOTT AND DEXUAN XIE, *Analysis of nonlinear multigrid methods with numerical integration*, to be submitted to Math. Comp., 1995.

[15] DEXUAN XIE, *New nonlinear multigrid analysis*, Chapter 6 of his Ph.D. Thesis, University of Houston, 1995.

[16] A. ŽENÍŠEK: *Nonlinear Elliptic and Evolution Problems and Their Finite Element Approximations,* Academic Press, 1990.

[17] JINCHAO XU: *Iterative Methods by Space Decomposition and Subspace Correction,* SIAM Review, Dec. 1992.

MULTIGRID METHOD FOR MODELING MULTI-DIMENSIONAL COMBUSTION WITH DETAILED CHEMISTRY

Xiaoqing Zheng, Chaoqun Liu, Changming Liao and Zhining Liu
Mathematics Department, University of Colorado at Denver
Denver, CO 80217

Steve McCormick
Program in Applied Mathematics, University of Colorado at Boulder
Boulder, CO 80309

SUMMARY

A highly accurate and efficient numerical method is developed for modeling 3-D reacting flows with detailed chemistry. A contravariant velocity-based governing system is developed for general curvilinear coordinates to maintain simplicity of the continuity equation and compactness of the discretization stencil. A fully-implicit backward Euler technique and a third-order monotone upwind-biased scheme on a staggered grid are used for the respective temporal and spatial terms. An efficient semi-coarsening multigrid method based on line-distributive relaxation is used as the flow solver. The species equations are solved in a fully coupled way and the chemical reaction source terms are treated implicitly. Example results are shown for a 3-D gas turbine combustor with strong swirling inflows.

INTRODUCTION

Combustion simulation generally requires the solution of the coupled equations of mass, momentum, species balance and energy with detailed thermodynamic and transport relations and finite-rate chemistry. In order to alleviate the strong interaction between the flow and combustion, and to avoid solving this huge system at the same time, the governing equations are usually solved in a semi-coupled way that the chemical reaction part and fluid flow part are treated separately. For the flow part, the mass, momentum and energy equations can be solved by using the existing CFD code; therefore, most efforts towards modeling combustion are concentrated on the reaction part. Many progresses have been made in solving the chemical species equations [1-8].

It is well realized that the reaction part, that involves multi-species, multi-step, finite rate kinetics, is a sensitive and stiff system, and it takes most of CPU time in most computations. Most of the successful combustion simulations are based on the coupled solution of chemical reaction system. There has not been found a general efficient way to decouple the system and reduce the cost in each iteration. Therefore the most effective approach is to reduce the iteration number. Since the flow field acts as the carrier of chemical reaction, it can be anticipated that a fast established flow field will provide a stable base for the reactions and therefore make the species equations easy to converge. As shown in our previous work [9,7,8], very efficient CFD methods will greatly reduce the iteration numbers of the reaction part which is very costly. Furthermore, for practical 3-D combustion, the flow field may be very complex, then the flow part could take considerable portion of the total CPU time. Therefore, the development of very efficient CFD methods and reaction modeling method is equally important in combustion simulations.

This paper describes a very accurate and efficient numerical method we have developed for calculating general 3-D reacting flows with detailed chemistry. The principal focus is put on the development of a high efficient solution method and high accurate scheme for chemical species transport equations. Based on the finite volume frame, an implicit method is developed to solve the 3-D Navier-Stokes equations and chemical species transport equations in general curvilinear coordinates. A distinctive feature of this method is that the contravariant velocities are employed as the dependent variables. The momentum equations of contravariant velocities are discretized in staggered control volumes while the energy equations and species equations are integrated basically by using a cell-centered finite volume scheme. In this way, the discretized mass equation remains its simple form as in the Cartesian grids and the stencil is spatially the most compact. A third-order monotone upwind-biased scheme by van Leer [10,11] is used for all the convection terms of flow equations and species equations to minimize numerical diffusion and maintain the sharp gradients present in flames.

This method was tested by applying to calculate the strong swirling combustion in a 3-D gas turbine combustor. For a 49x65x65 grid of 207,025 grid points, the calculation takes only about 200 time steps and 21.3 CRAY-YMP hours to reduce residuals by more than three orders of magnitude for all governing equations.

GOVERNING EQUATIONS

The governing equations for general compressible reacting flows in integration form can be summarized as follows.

Mass conservation:

$$\int_\Omega \frac{\partial \rho}{\partial t} d\Omega + \int_\Gamma \rho\vec{q}\cdot\vec{n} ds = 0 \tag{1}$$

Momentum conservation:

$$\int_\Omega \frac{\partial \rho\vec{q}}{\partial t} d\Omega + \int_\Gamma \rho\vec{q}(\vec{q}\cdot\vec{n}) ds = \int_\Gamma \vec{\tau_n} ds \tag{2}$$

In low speed combustion, the kinetic energy is negligible comparing with enthalpy; therefore, the energy conservation can be simplified as [12]:

$$\int_\Omega \frac{\partial(\rho h - p)}{\partial t} d\Omega + \int_\Gamma \rho h(\vec{q}\cdot\vec{n}) ds = \int_\Gamma \vec{\tau_n}\cdot\vec{q} ds + \int_\Gamma \lambda_h(\nabla h\cdot\vec{n}) ds \tag{3}$$

Chemical species equation:

$$\int_\Omega \frac{\partial \rho Y_\alpha}{\partial t} d\Omega + \int_\Gamma \rho Y_\alpha(\vec{q}\cdot\vec{n}) ds = \int_\Gamma \lambda_Y(\nabla Y_\alpha\cdot\vec{n}) ds + \int_\Omega R_\alpha d\Omega \tag{4}$$

$$\alpha = 1, 2, \cdots, NS,$$

Enthalpy and state equations:

$$h = h(Y_\alpha, T), \quad p = \sum_\alpha \frac{Y_\alpha}{W_\alpha}\rho R T. \tag{5}$$

where t is time, Ω is a fixed control volume with boundary Γ, ρ is density, p is pressure, $\vec{q}$ is velocity vector, T is the temperature, h is the enthalpy, $\vec{n}$ is the unit outer normal vector of the boundary, τ_n is the total viscous stress acted on a surface with outer normal vector $\vec{n}$, and R_α is the chemical reaction rate of species α. R, Y_α, and W_α are the gas constant, the mass fraction and molecular weight of species α, respectively, and the specific enthalpy and species diffusion coefficients are determined from

$$\lambda_h = \left(\frac{\mu_C}{Pr_L} + \frac{\mu_T}{Pr_T}\right), \quad \lambda_Y = \left(\frac{\mu_C}{Sc_L} + \frac{\mu_T}{Sc_T}\right)$$

where μ_C is molecular viscosity, μ_T the turbulent viscosity determined from turbulence model, Pr_L and Pr_T are the laminar and turbulent Prandtl numbers, and Sc_L and Sc_T are the laminar and turbulent Schmidt numbers, respectively. From the constitutive relations, we have:

$$[\tau] = -(p + \frac{2}{3}\mu\nabla\cdot\vec{q})[I] + 2\mu[\varepsilon] \tag{6}$$

$$\varepsilon_{i,j} = \left[\frac{\partial q_i}{\partial x_j} + \frac{\partial q_j}{\partial x_i}\right] \tag{7}$$

$$\mu = \mu_C + \mu_T \tag{8}$$

The enthalpy h and molecular viscosity μ can be calculated by the following formulas:

$$\begin{aligned} h &= \sum_\alpha Y_\alpha h_\alpha, \\ h_\alpha &= \int_0^T C_{P_\alpha} dT_\alpha = h_{0_\alpha} + \int_{T_0}^T C_{P_\alpha} dT_\alpha, \\ C_{P_\alpha} &= C_{P_\alpha}^0 + C_{P_\alpha}^1 T + C_{P_\alpha}^2 T^2 + C_{P_\alpha}^3 T^3 + C_{P_\alpha}^4 T^4, \\ \mu_C &= \sum_\alpha Y_\alpha \mu_\alpha, \\ \mu_\alpha &= \mu_\alpha^0 + \mu_\alpha^1 T + \mu_\alpha^2 T^2 + \mu_\alpha^3 T^3 + \mu_\alpha^4 T^4. \end{aligned} \tag{9}$$

where h_{0_α} is the standard formation enthalpy of αth species, $C_{P_\alpha}^0, C_{P_\alpha}^1, \cdots, C_{P_\alpha}^4$, $\mu_\alpha^0, \mu_\alpha^1, \cdots, \mu_\alpha^4$ are polynomial coefficients for C_{P_α} and μ_α, respectively.

All thermal and transport parameters are obtained by linking with CHEMKIN-II [13] standard libraries.

CHEMICAL REACTION MODEL

For laminar flames, the chemical reaction rate R_α for the αth species can be calculated by

$$R_\alpha = \omega_\alpha \sum_{j=1}^{N_R} [(\nu_{j\alpha}^P - \nu_{j\alpha}^R)(K_j^f \prod_{l=1}^{N_S} n_l^{\nu_{jl}^R} - K_j^b \prod_{l=1}^{N_S} n_l^{\nu_{jl}^P})] \tag{10}$$

where ω_α is the molecular weight of species α, N_R is the total number of reaction steps, N_S is the total number of species, $\nu_{j\alpha}^P$ ($\nu_{j\alpha}^R$) refers to the stoichiometric coefficient of products (reactants), and $n_l = \frac{\rho Y_l}{\omega_l}$.

The function K_j^f (K_j^b) is the rate constant for the forward (backward) reaction step j. We assume K_j^f has the following Arrhenius temperature dependent form:

$$K_j^f = A_j^f T_j^{\alpha_j^f} exp(-\frac{E_j^f}{RT}), \tag{11}$$

and K_j^b has a similar expression. The reverse rate constant can be written in terms of the forward rate constant and the equilibrium constant K_j^c as

$$K_j^b = K_j^f / K_j^c. \tag{12}$$

Here, K_j^c are also obtained by calling CHEMKIN-II. The pre-exponential factor A_j^f, the temperature exponent α_j^f, and the activation energy E_j^f can be compiled from published experimental work.

For turbulent reacting flows, the Algebraic Correlation Closure(ACC) model is used to introduce a correction term to the reaction rate [7,8].

CONTRAVARIANT VELOCITIES AND STAGGERED GRID

One may think use of contravariant velocity on staggered grid will result messy governing equations and cause great difficulties in coding. However, that is not always true. Following are the reasons why we choose it to solve reacting flows on arbitrary grid:

- Using staggered grid can result more accurate and robust schemes as concluded by numerical analysis and confirmed in previous calculations on regular Cartesian grids.
- On general curvilinear grids, staggered grid method can be made of best use by combining with contravariant velocities. For each contravariant velocity component, the discretization stencil for its main direction pressure gradient is spatially the most compact, therefore eliminating the possibility of odd-even decoupling of pressure.
- The use of the contravariant velocity also benefits the solution of mass, energy and chemical species equations. The flow convection can be accurately represented.
- With use of proper discretization method and careful selection of definition locations of variables, the governing equations can be kept simple enough for the momentum equations, and even simpler for all scalar conservation equations.
- Most importantly, this method will retain the close relation between mass flux and pressure difference on curvilinear grids. Therefore the pressure-correction method can be used very efficiently. This feature yields a fast convergence rate on curvilinear grids which is similar to that on Cartesian grids.

Let (u,v,w) be the velocity components in Cartesian coordinates (x,y,z), and $(\bar{U}, \bar{V}, \bar{W})$ be the contravariant velocity under computational coordinates (ξ, η, ζ); their relations can be described as:

$$\left.\begin{array}{l} \bar{U} = J(u\xi_x + v\xi_y + w\xi_z) \\ \bar{V} = J(u\eta_x + v\eta_y + w\eta_z) \\ \bar{W} = J(u\zeta_x + v\zeta_y + w\zeta_z) \end{array}\right\} \tag{13}$$

where J is the transformation Jacobian from (x,y,z) to (ξ, η, ζ).

From the above relations, the velocity components in x,y,z direction can be found:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = A \begin{bmatrix} \bar{U} \\ \bar{V} \\ \bar{W} \end{bmatrix}, \quad A = \begin{bmatrix} J\xi_x & J\xi_y & J\xi_z \\ J\eta_x & J\eta_y & J\eta_z \\ J\zeta_x & J\zeta_y & J\zeta_z \end{bmatrix}^{-1} \tag{14}$$

Equation 14 will be frequently used hereafter; for simplicity it is denoted as:

$$q_l = a_{lm} U^m \tag{15}$$

where $[q_1, q_2, q_3]^T = [u, v, w]^T$ and $[U^1, U^2, U^3]^T = [\bar{U}, \bar{V}, \bar{W}]^T$.

In this work, the basic scheme is the finite volume method. The computational domain is discretized into a number of quadrilateral cells in two dimensions or hexahedral cells in three dimensions. As in Fig. 1, 1-2-3-4-5-6-7-8 forms a typical cell in three dimensional problems. In finite volume formulation, the contravariant velocities can be expressed as:

$$\begin{array}{l} \bar{U}_{i+\frac{1}{2},j,k} = (\vec{q} \cdot \vec{S}_{5678})_{i+\frac{1}{2},j,k} \\ \bar{V}_{i,j+\frac{1}{2},k} = (\vec{q} \cdot \vec{S}_{2376})_{i,j+\frac{1}{2},k} \\ \bar{W}_{i,j,k+\frac{1}{2}} = (\vec{q} \cdot \vec{S}_{3487})_{i,j,k+\frac{1}{2}} \end{array} \tag{16}$$

where subscripts i, j, k denote the cell index in each of the three curvilinear coordinate directions, respectively. In order to retain the merit of staggered grid, the contravariant velocities are defined at different locations as shown in Eqn. 16 and Fig. 1.

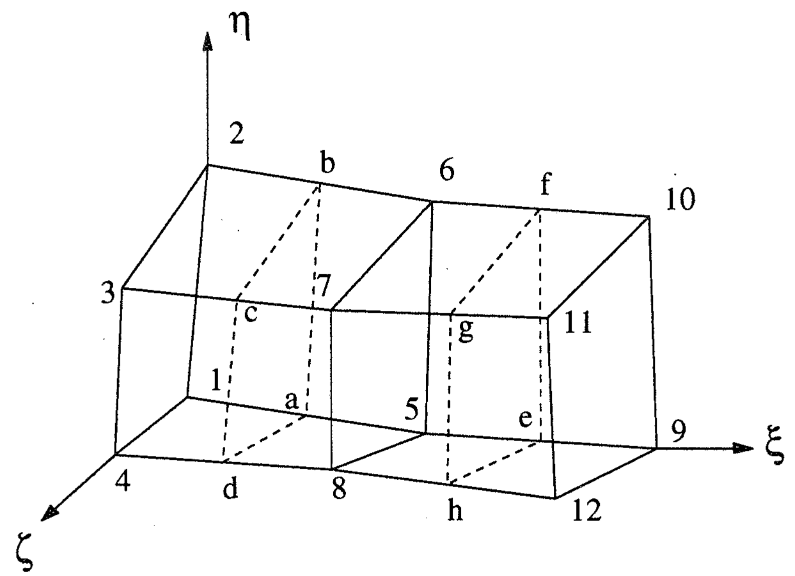


Figure 1: Cell Locations in Three Dimensional Grid

Generally the face vectors are denoted as following for clarity:

$$\begin{aligned} \vec{S}^1 &= \vec{S}_{\xi=const.} = (S^{1x}, S^{1y}, S^{1z}) \\ \vec{S}^2 &= \vec{S}_{\eta=const.} = (S^{2x}, S^{2y}, S^{2z}) \\ \vec{S}^3 &= \vec{S}_{\zeta=const.} = (S^{3x}, S^{3y}, S^{3z}) \end{aligned} \tag{17}$$

In the finite volume frame, equation 13 and equation 14 are expressed as:

$$\left.\begin{aligned} \bar{U} &= uS^{1x} + vS^{1y} + wS^{1z} \\ \bar{V} &= uS^{2x} + vS^{2y} + wS^{2z} \\ \bar{W} &= uS^{3x} + vS^{3y} + wS^{3z} \end{aligned}\right\} \tag{18}$$

and

$$A = \begin{bmatrix} S^{1x} & S^{1y} & S^{1z} \\ S^{2x} & S^{2y} & S^{2z} \\ S^{3x} & S^{3y} & S^{3z} \end{bmatrix}^{-1} \tag{19}$$

In the actual computation, $\rho\bar{U}$, $\rho\bar{V}$, and $\rho\bar{W}$ are regarded as the dependent variables instead of $\bar{U}$, $\bar{V}$, and $\bar{W}$, because they are conserved quantities and the resulting governing equations are relatively simple. Their definition locations are the same as those of $\bar{U}$, $\bar{V}$, and $\bar{W}$. $\rho\bar{U}$ is defined at $(i+\frac{1}{2}, j, k)$, $\rho\bar{V}$ is defined at $(i, j+\frac{1}{2}, k)$, and $\rho\bar{W}$ is defined at $(i, j, k+\frac{1}{2})$. All other variables, ρ, p, h, and Y_α, are defined at the cell centers. Only ρ, $\rho\bar{U}$, $\rho\bar{V}$, $\rho\bar{W}$, h, and Y_α are the dependent variables which are solved directly from the integral conservation equations (1-4). All other parameters are determined from the relations (5-10).

The governing equations for contravariant velocities can be established through coordinate transformation, then their forms are indeed quite complicated. Actually we can find an easy way to obtain the equations by applying the momentum equation to certain control volumes. For example, the equation for $\rho\bar{U}_{i+\frac{1}{2},j,k}$ can be obtained by simply multiplying the Eqn. 2 with the face vector $\vec{S}^1_{i+\frac{1}{2},j,k}$, then applied to control volume $Vol_{i+\frac{1}{2},j,k}$, which is formed by connecting ξ-line mid-points a-b-c-d-h-e-f-g as shown in Fig. 1

$$\int_{\Omega_{\bar{U}}} \frac{\partial \rho(\vec{q}\cdot\vec{S}^1)}{\partial t} d\Omega + \int_{\Gamma_{\bar{U}}} \rho(\vec{q}\cdot\vec{S}^1)(\vec{q}\cdot\vec{n})\, ds = \int_{\Gamma_{\bar{U}}} \vec{S}^1 \cdot \vec{\tau}_n ds \tag{20}$$

Notice $\vec{S}^1 = \vec{S}^1_{i+\frac{1}{2},j,k}$ is a constant vector within the control volume $\Omega_{\bar{U}} = Vol_{i+\frac{1}{2},j,k}$. In the above, all $\vec{q}$ will be eventually expressed in terms of $\bar{U}, \bar{V}$ and $\bar{W}$ by using Eqns. (18,19). We prefer to do the transformation later in the succeeding sections, because it will be much easier to do that after discretization.

The momentum equations for $\rho\bar{V}$ and $\rho\bar{W}$ can be obtained in the similar way by applying to $Vol_{i,j+\frac{1}{2},k}$ and $Vol_{i,j,k+\frac{1}{2}}$, respectively.

All the other equations, i.e., mass conservation, energy conservation and species equations, are applied to control volume $Vol_{i,j,k}$. They can be put in a general form:

$$\int_\Omega \frac{\partial \rho\phi}{\partial t} d\Omega + \int_\Gamma \rho\phi(\vec{q}\cdot\vec{n})ds = \int_\Gamma \lambda(\nabla\phi\cdot\vec{n})ds + \int_\Gamma Fds + \int_\Omega Sds \tag{21}$$

where $\phi = [1, h, Y_\alpha]^T, F = [0, \vec{\tau_n}\cdot\vec{q}, 0]^T$ and $S = [0, 0, R_\alpha]^T$ with $\alpha = 1, 2, \cdots, NS$.

The above equations are not their final forms; the Cartesian velocity $\vec{q}$ is still used for simplicity. It will be replaced by contravariant velocity during the discretization process in the next two sections.

STAGGERED FINITE VOLUME SCHEME

In this section, we begin to discretize the governing equations described in the last section.

Momentum Equations

The $\rho\bar{U}$-equation (20) is applied to the staggered control volume $Vol_{i+\frac{1}{2},j,k}$, which is discretized by using finite volume method as

$$Vol_{i+\frac{1}{2},j,k}\frac{(\rho\bar{U})^{n+1}_{i+\frac{1}{2},j,k} - (\rho\bar{U})^{n}_{i+\frac{1}{2},j,k}}{\Delta t} + \sum_{l=1}^{6}([\vec{S}^1\cdot(\rho\vec{q})_l](\vec{q}\cdot\vec{S})_l = Vis_{\rho\bar{U}} \tag{22}$$

where l is the cell surface index, ranges all the 6 cell surfaces of the control volume $Vol_{i+\frac{1}{2},j,k}$. $Vis_{\rho\bar{U}}$ is the total viscous stress component in $\vec{S}^1_{i+\frac{1}{2},j,k}$ direction acted on the surface of control volume $Vol_{i+\frac{1}{2},j,k}$. It will be described in the next section.

Based on the idea of MUSCL scheme by Van Leer [10,11], a partially upwind-biased scheme is developed to approximate the momentum fluxes through cell surfaces. The basic idea is that the flux through the control volume surface is regarded as the product of the mass flow and the conserved quantity. According to the sign of mass flux, the conserved quantity is set to its upwind-side value. Thanks to the staggered scheme, the mass flux through the surfaces is always directly available. There are only two possible locations for all the control volume surfaces, either the surface lies along with one of the original grid surfaces or it runs through the original grid cell center. In the former case, the mass flux is already defined there. In the latter case, since the Cartesian velocity and density are defined at the cell center, the mass flow also can be found straightforwardly. Therefore only the conserved quantity at the surface is needed to be interpolated or obtained through reconstruction of data from the cell-averaged values like Van Leer's MUSCL method. This feature ensures that the calculated flux is continuous when mass flow changes sign. For example, if the flux $(\vec{F})$ through a control volume surface $(\vec{S})$ in i direction is consisted of mass flow $(\vec{M})$ and the conserved quantity (ψ), then

$$\vec{F}_i = (\vec{M}\cdot\vec{S})_i \;\; \psi_i = (\vec{M}\cdot\vec{S})^+_i \;\; \psi_{i(-)} + (\vec{M}\cdot\vec{S})^-_i \;\; \psi_{i(+)} \tag{23}$$

In the above, the superscripts +,- on a variable denote the positive and negative part of the variable, respectively,

$$M^+ = max(M, 0), \quad M^- = min(M, 0) \tag{24}$$

and the superscripts (+),(-) on an index indicate that the variable is taking the limit value on the interface from the left or the right, respectively. For instance, in i direction we have:

$$\psi_{i(-)} = \lim_{\substack{l \leq i \\ l \to i}} \psi_l, \quad \psi_{i(+)} = \lim_{\substack{l \geq i \\ l \to i}} \psi_l \tag{25}$$

High-resolution schemes up to third order can be constructed by setting

$$\psi_{i(-)} = \psi_{i-1/2} + \frac{\sigma^{\psi}_{i-\frac{1}{2}}}{4}[(1-\kappa)\nabla + (1+\kappa)\Delta]\psi_{i-1/2} \tag{26}$$

$$\psi_{i(+)} = \psi_{i+1/2} - \frac{\sigma^{\psi}_{i+\frac{1}{2}}}{4}[(1+\kappa)\nabla + (1-\kappa)\Delta]\psi_{i+1/2} \tag{27}$$

where ∇ and Δ are backward and forward difference operators, and κ is a parameter used to control the order of the scheme. $\kappa = (1/3)$ is used in the present method to construct the third-order scheme. When $\kappa = -1$ the scheme reduces to the second-order fully upwind method. The limiter σ is adopted to ensure the monotone interpolation following Koren [14] as:

$$\sigma^{\psi}_{l-\frac{1}{2}} = \frac{3\nabla\psi_{l-\frac{1}{2}}\Delta\psi_{l-\frac{1}{2}} + \theta}{2(\nabla\psi_{l-\frac{1}{2}} - \Delta\psi_{l-\frac{1}{2}})^2 + 3\nabla\psi_{l-\frac{1}{2}}\Delta\psi_{l-\frac{1}{2}} + \theta} \tag{28}$$

where θ, a small constant with a typical value of 10^{-20}, is added to prevent division by zero.

In our solution algorithm, only $(\rho\bar{U})_{i+\frac{3}{2},j,k}$,$(\rho\bar{U})_{i-\frac{1}{2},j,k}$, $(\rho\bar{U})_{i+\frac{1}{2},j+1,k}$, $(\rho\bar{U})_{i+\frac{1}{2},j-1,k}$, $(\rho\bar{U})_{i+\frac{1}{2},j,k+1}$ $(\rho\bar{U})_{i+\frac{1}{2},j,k-1}$, $(\rho\bar{U})_{i+\frac{1}{2},j,k}$, $p_{i,j,k}$ and $p_{i+1,j,k}$ are treated implicitly for $\rho\bar{U}$-equation. In general, the $\rho\bar{U}$-equation can be expressed in δ form as:

$$\begin{aligned}
& A_E\delta(\rho\bar{U})_{i+\frac{3}{2},j,k} + A_W\delta(\rho\bar{U})_{i-\frac{1}{2},j,k} + A_N\delta(\rho\bar{U})_{i+\frac{1}{2},j+1,k} + A_S\delta(\rho\bar{U})_{i+\frac{1}{2},j-1,k} \\
+ \quad & A_F\delta(\rho\bar{U})_{i+\frac{1}{2},j,k+1} + A_B\delta(\rho\bar{U})_{i+\frac{1}{2},j,k-1} + A_C\delta(\rho\bar{U})_{i+\frac{1}{2},j,k} \\
+ \quad & A^p_L\delta p_{i,j,k} + A^p_R\delta p_{i+1,j,k} = -Ru_{i+\frac{1}{2},j,k}
\end{aligned} \tag{29}$$

where Ru denotes the residual of $\rho\bar{U}$-equation, including convection and diffusion part.

Similarly, the momentum equations of $\rho\bar{V}$ and $\rho\bar{W}$ can be found.

Scalar Conservation Equations

All the scalar conservation equations (21) are applied to control volume $Vol_{i,j,k}$ with cell-centered finite volume scheme. The above-used upwind-biased scheme with limiter are used for the convection terms, second order compact central difference scheme for the diffusion terms. The only exception is the mass conservation equation, which benefits most from the staggered grid, the discretized equation has the simplest form and is the most compact in space in terms of contravariant flux velocity

$$\begin{aligned}
& \delta(\rho\bar{U})_{i+\frac{1}{2},j,k} - \delta(\rho\bar{U})_{i-\frac{1}{2},j,k} + \delta(\rho\bar{V})_{i,j+\frac{1}{2},k} - \delta(\rho\bar{V})_{i,j-\frac{1}{2},k} \\
+ \quad & \delta(\rho\bar{W})_{i,j,k+\frac{1}{2}} - \delta(\rho\bar{W})_{i,j,k-\frac{1}{2}} = -Rm_{i,j,k}
\end{aligned} \tag{30}$$

where

$$\begin{aligned}
Rm_{i,j,k} \quad = \quad & Vol\frac{\rho^{n+1}_{i,j,k} - \rho^{n}_{i,j,k}}{\Delta t} + (\rho\bar{U})^n_{i+\frac{1}{2},j,k} - (\rho\bar{U})^n_{i-\frac{1}{2},j,k} \\
+ \quad & (\rho\bar{V})^n_{i,j+\frac{1}{2},k} - (\rho\bar{V})^n_{i,j-\frac{1}{2},k} + (\rho\bar{W})^n_{i,j,k+\frac{1}{2}} - (\rho\bar{W})^n_{i,j,k+\frac{1}{2}}
\end{aligned} \tag{31}$$

In our solution method the time-dependent term of mass equation is dropped for fast convergence.

All other equations are discussed here in their general form (21) except for the source term and the stress work term. The source terms of the species equations are usually dominant and of strong non-linearity. We will discuss the treatment of those source terms in the next sub-section. The stress work term in the energy equation will be discussed in the next section, since it has no contribution to the implicit coefficients. If we leave the implicit coefficients contributed by the source terms in the next sub-section, the discretized forms of Eqn.(21) are assumed to have the following form:

$$\begin{array}{l}\Phi_E\delta\phi_{i+1,j,k} + \Phi_W\delta\phi_{i-1,j,k} + \Phi_N\delta\phi_{i,j+1,k} + \Phi_S\delta\phi_{i,j-1,k} \\ +\Phi_F\delta\phi_{i,j,k+1} + \Phi_B\delta\phi_{i,j,k-1} + \Phi_C\delta\phi_{i,j,k} = -Res(\phi)\end{array} \tag{32}$$

where $Res(\phi)$ is the residual of ϕ-equation.

The convection term is discretized by using the same method described in last sub-section for convection terms of momentum equations. The diffusion term on the right side of Eqn.(21) is discretized through two steps. First we calculate the gradient $\nabla\phi$ on the cell surface by applying Gauss's formula to locally-formed staggered control volume, then assemble the integration. Since the gradients are computed locally, the resulted scheme reduces to a compact one when regular grid is used.

Implicit Treatment of Reaction Source Term

The major difficulty in calculation of finite rate combustion is the stiffness of the species equations. To solve this problem, the source terms (production rate of chemical reaction) must be treated implicitly.

In the last subsection, the discretization of time dependent, convection and diffusion terms of the general scalar conservation equation is discussed. For the chemical species equations, the discretized equations can be written as:

$$\begin{array}{l}\Phi_E\delta Y_{\alpha_{i+1,j,k}} + \Phi_W\delta Y_{\alpha_{i-1,j,k}} + \Phi_N\delta Y_{\alpha_{i,j+1,k}} + \Phi_S\delta Y_{\alpha_{i,j-1,k}} \\ +\Phi_F\delta Y_{\alpha_{i,j,k+1}} + \Phi_B\delta Y_{\alpha_{i,j,k-1}} + \Phi_C\delta Y_{\alpha_{i,j,k}} = -\left[C_T(Y_\alpha)^n - D_T(Y_\alpha)^n - R_\alpha\right]\end{array} \tag{33}$$

where $\delta(\ ) = (\ )^{n+1} - (\ )^n$, C_T is the convection term and D_T the diffusion term. R_α is the reaction rate defined in Eqn.(10)

$$R_\alpha \;=\; W_\alpha \sum_{m=1}^{N_R}[(\nu^P_{m\alpha} - \nu^R_{m\alpha})(K^f_m \prod_{l=1}^{N_S}(\frac{\rho Y_l}{W_l})\nu^R_{ml} - K^b_m \prod_{l=1}^{N_S}(\frac{\rho Y_l}{W_l})\nu^P_{ml}] \tag{34}$$

The reaction rate is usually very large and dominant near the flame front. Therefore, implicit treatment for the production rate term is necessary. Using Taylor expansion, we have

$$R^{n+1}_\alpha = R^n_\alpha + \sum_m \frac{\partial R_\alpha}{\partial Y_m}\delta Y_m + \sum_m O(\delta Y_m^2). \tag{35}$$

By defining

$$\mathbf{R} = (R_1, R_2, \cdots, R_{N_S})^T,\ \delta\mathbf{Y} = (\delta Y_1, \delta Y_2, \cdots, \delta Y_{N_S})^T,\ \mathbf{D}_{\alpha m} = \frac{\partial R_\alpha}{\partial Y_m}, \tag{36}$$

we may have

$$\mathbf{R}^{n+1} \approx \mathbf{R}^n + \mathbf{D}\delta\mathbf{Y}. \tag{37}$$

where $\mathbf{D}$ is a N_S by N_S matrix.

It is apparent that the implicit treatment of R_α requires the coupled solution of the species equations. By denoting $Res_\alpha = C_T(Y_\alpha)^n - D_T(Y_\alpha)^n - R^n_\alpha$, the residual of αth species equation and $\mathbf{Res} = (Res_1, Res_2, \cdots, Res_{N_S})^T$, the residual vector, then Eqn.(33) becomes

$$\begin{array}{l}\Phi_E\mathbf{I}\delta\mathbf{Y}_{i+1,j,k} + \Phi_W\mathbf{I}\delta\mathbf{Y}_{i-1,j,k} + \Phi_N\mathbf{I}\delta\mathbf{Y}_{i,j+1,k} + \Phi_S\mathbf{I}\delta\mathbf{Y}_{i,j-1,k} \\ +\Phi_F\mathbf{I}\delta\mathbf{Y}_{i,j,k+1} + \Phi_B\mathbf{I}\delta\mathbf{Y}_{i,j,k-1} + (\Phi_C\mathbf{I} + \mathbf{D})\delta\mathbf{Y}_{i,j,k} = -\mathbf{Res}\end{array} \tag{38}$$

where **I** is a unit matrix with the elements

$$I_{lm} = \begin{cases} 0 & \text{if } l \neq m \\ 1 & \text{if } l = m \end{cases} \tag{39}$$

and Φ is a scalar.

Eqn.(38) is the final form of the species equations. They are solved in a coupled way. If line-relaxation is used along j-line and Gauss-Seidel iteration used in i, k directions, for instance, then the equation (38) is rewritten as

$$\begin{aligned} &\Phi_S \mathbf{I} \delta \mathbf{Y}^{new}_{i,j-1,k} + (\Phi_C \mathbf{I} + \mathbf{D}) \delta \mathbf{Y}^{new}_{i,j,k} + \Phi_N \mathbf{I} \delta \mathbf{Y}^{new}_{i,j+1,k} \\ &= -\mathbf{Res} - \Phi_E \mathbf{I} \delta \mathbf{Y}^{old}_{i+1,j,k} - \Phi_W \mathbf{I} \delta \mathbf{Y}^{new}_{i-1,j,k} - \Phi_F \mathbf{I} \delta \mathbf{Y}^{old}_{i,j,k+1} - \Phi_B \mathbf{I} \delta \mathbf{Y}^{new}_{i,j,k-1} \end{aligned} \tag{40}$$

The left side of above equation forms a block-tridiagonal system, which can be solved by using the tailor-made algorithm combined with a Gauss Elimination method for the small block matrix inversion.

VISCOUS STRESS

Generally the viscous stress acted on a surface $\vec{S} = \{S_x, S_y, S_z\}$ with outer normal $\vec{n} = \frac{\vec{S}}{S}$ is defined as:

$$\begin{aligned} \vec{\tau}_n S &= \vec{\tau}_x S_x + \vec{\tau}_y S_y + \vec{\tau}_z S_z \\ &= \vec{i}(\tau_{xx} S_x + \tau_{yx} S_y + \tau_{zx} S_z) + \vec{j}(\tau_{xy} S_x + \tau_{yy} S_y + \tau_{zy} S_z) + \vec{k}(\tau_{xz} S_x + \tau_{yz} S_y + \tau_{zz} S_z) \end{aligned} \tag{41}$$

where $\vec{i}, \vec{j}, \vec{k}$ is the unit vector in x, y, z direction, respectively.

The viscous force component in $\vec{n}_u$ direction acted on $\vec{S}$ surface can be obtained by multiplication of the above equation with $\vec{n}_u$:

$$\begin{aligned} (\vec{\tau}_n \cdot \vec{n}_u) S &= n_{ux}(\tau_{xx} S_x + \tau_{yx} S_y + \tau_{zx} S_z) + n_{uy}(\tau_{xy} S_x + \tau_{yy} S_y + \tau_{zy} S_z) \\ &+ n_{uz}(\tau_{xz} S_x + \tau_{yz} S_y + \tau_{zz} S_z) \end{aligned} \tag{42}$$

From Eqn.(5), we have

$$\tau_{lm} = \begin{cases} \mu(\frac{\partial u^l}{\partial x^m} + \frac{\partial u^m}{\partial x^l}) & l \neq m \\ -(p + \frac{2}{3}\mu \nabla \cdot \vec{q}) + 2\mu \frac{\partial u^l}{\partial x^m} & l = m \end{cases} = \begin{cases} B^{lm} & l \neq m \\ -p + \frac{2}{3} B^{mm} & l = m \end{cases} \tag{43}$$

where $B^{lm}_{i,j,k} = \mu_{i,j,k} \left(\frac{\partial u^l}{\partial x^m} + \frac{\partial u^m}{\partial x^l} \right)_{i,j,k}$.

In the finite volume formulation, velocity strain can be calculated as:

$$\begin{aligned} \left. \frac{\partial u^l}{\partial x^m} \right|_{i,j,k} = \frac{1}{Vol_{i,j,k}} \quad [\quad & (u^l S^{1m})_{i+\frac{1}{2},j,k} - (u^l S^{1m})_{i-\frac{1}{2},j,k} \\ + \quad & (u^l S^{2m})_{i,j+\frac{1}{2},k} - (u^l S^{2m})_{i,j-\frac{1}{2},k} + (u^l S^{3m})_{i,j,k+\frac{1}{2}} - (u^l S^{3m})_{i,j,k-\frac{1}{2}}] \end{aligned} \tag{44}$$

Hereafter, all subscripts, except those indicating grid location, are placed upper-right like superscripts to avoid confusing with cell index.

After substituting the above equation into Eqn.(42), we have:

$$(\vec{\tau}_n \cdot \vec{n}_u) S = n^m_u \tau^{lm} S^l = \left(1 - \frac{\delta^{lm}}{3} \right) n^m_u B^{lm} S^l - \delta^{lm} n^m_u p S^l \tag{45}$$

By introducing difference operator

$$\begin{cases} \delta_\xi(\ )_{i,j,k} = \delta_1(\ )_{i,j,k} = (\ )_{i+\frac{1}{2},j,k} - (\ )_{i-\frac{1}{2},j,k} \\ \delta_\eta(\ )_{i,j,k} = \delta_2(\ )_{i,j,k} = (\ )_{i,j+\frac{1}{2},k} - (\ )_{i,j-\frac{1}{2},k} \\ \delta_\zeta(\ )_{i,j,k} = \delta_3(\ )_{i,j,k} = (\ )_{i,j,k+\frac{1}{2}} - (\ )_{i,j,k-\frac{1}{2}} \end{cases} \tag{46}$$

and using Eqn.(15), B^{lm} then can be expressed in terms of contravariant velocity as:

$$B^{lm}_{i,j,k} = \left(\frac{\mu}{Vol}\right)_{i,j,k} \delta_n \left[(S^{nm}a^{lr} + S^{nl}a^{mr})U^r\right]_{i,j,k} \tag{47}$$

The viscous term in Eqn.(22) can be obtained by applying the above equation to each surface of the control volume $Vol_{i+\frac{1}{2},j,k}$, after substituting $\vec{n}_u$ in the above equation with $\vec{S}^1_{i+\frac{1}{2},j,k}$,

$$\begin{aligned} Vis(\bar{U}) &= \vec{S}^1_{i+\frac{1}{2},j,k} \cdot \{(S^1\vec{\tau}_{n_{s1}})_{i+1,j,k} - (S^1\vec{\tau}_{n_{s1}})_{i,j,k} + (S^2\vec{\tau}_{n_{s2}})_{i+\frac{1}{2},j+\frac{1}{2},k} - (S^2\vec{\tau}_{n_{s2}})_{i+\frac{1}{2},j-\frac{1}{2},k} \\ &\quad +(S^3\vec{\tau}_{n_{s3}})_{i+\frac{1}{2},j,k+\frac{1}{2}} - (S^3\vec{\tau}_{n_{s3}})_{i+\frac{1}{2},j,k-\frac{1}{2}}\Big\} \\ &= S^{1m}_{i+\frac{1}{2},j,k}\delta_n(pS^{nm})_{i+\frac{1}{2},j,k} \\ &+ \left(1-\frac{\delta^{lm}}{3}\right) S^{1m}_{i+\frac{1}{2},j,k}\delta_1(B^{lm}S^{1l})_{i+\frac{1}{2},j,k} + \delta_2(B^{lm}S^{2l})_{i+\frac{1}{2},j,k} + \delta_3(B^{lm}S^{3l})_{i+\frac{1}{2},j,k} \end{aligned} \tag{48}$$

Similarly, we can find the viscous terms in $\bar{V}$- and $\bar{W}$-equations.

In the energy equation, the viscous stress work is:

$$\begin{aligned} \int_\Gamma \vec{\tau}_n \cdot \vec{q} ds &= \Big\{(S^1\vec{\tau}_{n_{s1}} \cdot \vec{q})_{i+\frac{1}{2},j,k} - (S^1\vec{\tau}_{n_{s1}} \cdot \vec{q})_{i-\frac{1}{2},j,k} + (S^2\vec{\tau}_{n_{s2}} \cdot \vec{q})_{i,j+\frac{1}{2},k} \\ &\quad -(S^2\vec{\tau}_{n_{s2}} \cdot \vec{q})_{i,j-\frac{1}{2},k} + (S^3\vec{\tau}_{n_{s3}} \cdot \vec{q})_{i,j,k+\frac{1}{2}} - (S^3\vec{\tau}_{n_{s3}} \cdot \vec{q})_{i,j,k-\frac{1}{2}}\Big\} \\ &= \delta_n(pq^mS^{nm})_{i,j,k} \\ &+ \left(1-\frac{\delta^{lm}}{3}\right) \delta_1(B^{lm}q^mS^{1l})_{i,j,k} + \delta_2(B^{lm}q^mS^{2l})_{i,j,k} + \delta_3(B^{lm}q^mS^{3l})_{i,j,k} \end{aligned} \tag{49}$$

SOLUTION PROCEDURE

To solve the governing equations discretized in foregoing sections, an implicit time-marching method has been developed. The governing equations are divided into two sets: the flow part and the chemical reaction part. They are solved alternately. Different solving techniques are applied to those two sets of equations. In the following, the numerical procedure is described in detail.

Provision of Reaction Mechanism

For a given combustion problem, the chemical reaction mechanism is needed to be prescribed besides the fuel, oxidizer and boundary conditions. The chemical reaction mechanism is usually obtained through experiment. In the numerical simulation, it is represented by the pre-exponential factor A^f_j, the temperature exponent α^f_j and the activation energy E^f_j of the chemical reaction equations. Those parameters and reaction equations are specified through an input data file "mech" provided by users in our code. In our test case involved methane-air reaction, the C_1-chain reaction mechanism in Table 1 given by Xu [5] is adopted, in which 16 species are involved in 45 steps reaction chain.

Thermal and transport parameters are obtained by calling CHEMKIN-II subroutines and data bases.

Table 1. C_1-Chain Methane-Air Reaction Mechanism. Rate coefficients: $K = AT^{\alpha}exp(-\frac{E}{RT})$, units: moles, cubic centimeters, seconds, Kelvins, calories

No.	reaction	A	α	E
1	$CH_3 + H \rightleftharpoons CH_4$	1.90E+36	-7.	9050.
2	$CH_4 + O_2 \rightleftharpoons CH_3 + HO_2$	7.90E+13	0.	56000.
3	$CH_4 + H \rightleftharpoons CH_3 + H_2$	2.20E+4	3.	8750.
4	$CH_4 + O \rightleftharpoons CH_3 + OH$	1.60E+6	2.36	7400.
5	$CH_4 + OH \rightleftharpoons CH_3 + H_2O$	1.60E+6	2.1	2460.
6	$CH_2O + OH \rightleftharpoons HCO + H_2O$	7.53E+12	0.	167.
7	$CH_2O + H \rightleftharpoons HCO + H_2$	3.31E+14	0.	10500.
8	$CH_2O + M \rightleftharpoons HCO + H + M$	3.31E+16	0.	81000.
9	$CH_2O + O \rightleftharpoons HCO + OH$	1.81E+13	0.	3082.
10	$HCO + OH \rightleftharpoons CO + H_2O$	5.00E+12	0.	0.
11	$HCO + M \rightleftharpoons H + CO + M$	1.60E+14	0.	14700.
12	$HCO + H \rightleftharpoons CO + H_2$	4.00E+13	0.	0.
13	$HCO + O \rightleftharpoons OH + CO$	1.00E+13	0.	0.
14	$HCO + O_2 \rightleftharpoons HO_2 + CO$	3.00E+12	0.	0.
15	$CO + O + M \rightleftharpoons CO_2 + M$	3.20E+13	0.	-4200.
16	$CO + OH \rightleftharpoons CO_2 + H$	1.51E+7	1.3	-758.
17	$CO + O_2 \rightleftharpoons CO_2 + O$	1.60E+13	0.	41000.
18	$CH_3 + O_2 \rightleftharpoons CH_3O + O$	7.00E+12	0.	25652.
19	$CH_3O + M \rightleftharpoons CH_2O + H + M$	2.40E+13	0.	28812.
20	$CH_3O + H \rightleftharpoons CH_2O + H_2$	2.00E+13	0.	0.
21	$CH_3O + OH \rightleftharpoons CH_2O + H_2O$	1.00E+13	0.	0.
22	$CH_3O + O \rightleftharpoons CH_2O + OH$	1.00E+13	0.	0.
23	$CH_3O + O_2 \rightleftharpoons CH_2O + HO_2$	6.30E+10	0.	2600.
24	$CH_3 + O_2 \rightleftharpoons CH_2O + OH$	5.20E+13	0.	34574.
25	$CH_3 + O \rightleftharpoons CH_2O + H$	6.80E+13	0.	0.
26	$CH_3 + OH \rightleftharpoons CH_2O + H_2$	7.50E+12	0.	0.
27	$HO_2 + CO \rightleftharpoons CO_2 + OH$	5.80E+13	0.	22934.
28	$H_2 + O_2 \rightleftharpoons 2OH$	1.70E+13	0.	47780.
29	$OH + H_2 \rightleftharpoons H_2O + H$	1.17E+9	1.3	3626.
30	$H + O_2 \rightleftharpoons OH + O$	2.20E+14	0.	16800.
31	$O + H_2 \rightleftharpoons OH + H$	1.80E+10	1.	8826.
32	$H + O_2 + M \rightleftharpoons HO_2 + M^a$	2.10E+18	-1.	0.
33	$H + O_2 + O_2 \rightleftharpoons HO_2 + O_2$	6.70E+19	-1.42	0.
34	$H + O_2 + N_2 \rightleftharpoons HO_2 + N_2$	6.70E+19	-1.42	0.
35	$OH + HO_2 \rightleftharpoons H_2O + O_2$	5.00E+13	0.	1000.
36	$H + HO_2 \rightleftharpoons 2OH$	2.50E+14	0.	1900.
37	$O + HO_2 \rightleftharpoons O_2 + OH$	4.80E+13	0.	1000.
38	$2OH \rightleftharpoons O + H_2O$	6.00E+8	1.3	0.
39	$H_2 + M \rightleftharpoons H + H + M^b$	2.23E+12	0.5	92600.
40	$O_2 + M \rightleftharpoons O + O + M$	1.85E+11	0.5	95560.
41	$H + OH + M \rightleftharpoons H_2O + M$	7.50E+23	-2.6	0.
42	$H + HO_2 \rightleftharpoons H_2 + O_2$	2.50E+13	0.	700.
43	$HO_2 + HO_2 \rightleftharpoons H_2O_2 + O_2$	2.00E+12	0.	0.
44	$H_2O_2 + M \rightleftharpoons OH + OH + M$	1.30E+17	0.	45500.
45	$H_2O_2 + OH \rightleftharpoons H_2O + HO_2$	1.00E+13	0.	1800.

Third body efficiency with respect to Ar:

[a] $H_2O = 21, H_2 = 3.3, CO = 2.0, CO_2 = 5.0, N_2 = O_2 = 0.$

[b] $H_2O = 6, H = 2, H_2 = 3$

Starting Estimate

The governing system is highly nonlinear and its solution requires a good starting estimate. Similar to the work by Xu et al [5], we use a solution of infinitely fast combustion [9] as our initial guess. In the infinitely fast kinetics, the fuel and the oxidizer are separated by a thin exothermic reaction zone. In this zone the fuel and oxidizer are in stoichiometric proportion and the temperature and products of combustion are maximized. This infinitely fast reaction solution not only provides a good initial guess, but also helps overcome the difficulty of ignition with finite-rate combustion.

Solution Method

A fully implicit time-stepping scheme is developed. In the laminar case, the system consists of 21 equations (if there are 16 species). In the turbulent case there will be 23 equations. They are solved in groups:
(a) $\rho\bar{U}, \rho\bar{V}, \rho\bar{W}$ and p by solving the mass and momentum equations
(b) k, ϵ, μ_t by solving the turbulence model in turbulent combustion case
(c) h, Y_α by solving the energy and species equations
and, finally, updating
(d) ρ, μ by calling CHEMIKIN-II

For the flow part, a line-distribution updating scheme [9,15] is used. To further accelerate the convergence, a semi-coarsening multigrid method is developed. Here we only point out the techniques we used for our specific applications. In our method, the density and pressure are defined at the cell centers and the contravariant velocities are defined at cell interfaces. The density and pressure are transferred from finer level by area weighting to coarser grid; the contravariant velocities in coarser grid are simply set to the sum of those at corresponding interfaces. The residuals on finer grid are restricted to coarser by adding up the corresponding part to the staggered stencils. After relaxation is completed on coarser grid, the corrections are fed back to finer grid by bilinear interpolation.

For the reaction part, the energy equation is solved together with the species equations. An implicit alternate line-relaxation method is used for the energy equation. The species equations are treated in a fully coupled way. The reaction source terms, which are non-linear and usually troublesome, are treated implicitly by linearization. The block-line tridiagonal solver combined with vectorized pivoting Gauss elimination is used, which was found very effective to handle the sensitivity and stiffness of the system.

The multigrid method is used only for momentum and continuity equations in this work. The other equations, such as energy equation, species equations and κ, ϵ equations, are solved on a single grid. Therefore, we cannot achieve full multigrid efficiency. However, the whole process for solving our system is still substantially accelerated.

BOUNDARY CONDITIONS

The boundary type usually encountered can be classified as inflow, outflow, solid wall, symmetrical (slip) and periodical. At the inflow boundary, the flow velocity, enthalpy, and chemical species are specified, but the pressure is extrapolated from the interior; then the density is found herefrom by using the state equation.

For the outflow boundary, the back pressure is prescribed and other variables are extrapolated from the interior.

For solid wall boundary, since ghost cell is always introduced, both slip (symmetrical) and non-slip conditions can be easily implemented with use of contravariant velocities. Take example of wall condition on a $j = constant$ plane. For non-slip condition, reverse reflection is applied to all the contravariant velocities associated with the ghost cell. For slip (symmetrical) boundary, the reverse reflection is only applied to $\bar{V}$, direct reflection is applied $\bar{U}$ and $\bar{V}$. In both cases, the contravariant velocity $\bar{V}$ lies on this $j = constant$ plane is always set to zero.

The periodical boundary is the simplest. All the values on ghost cell are taken directly from the corresponding cell of other side.

All the boundary conditions are treated fully implicitly through modification of the implicit coefficients of the discretized equations at the boundary points.

NUMERICAL RESULTS

This method was applied to calculate the strong swirling combustion in a 3-D gas turbine combustor. The computational conditions and grid information are summarized in Table 2.

Table 2 Strong Swirling Combustion in a 3-D Model Combustor

Table 2.1 Working Conditions

Inflow Speed	Fuel	Oxidizer	Species Number	Reaction Steps
0.0988(average), 30° swirling angle	Methane	Air	16	45 (Table 1)

Table 2.2 Summary of CPU Time and Convergence on Different Grids

	Iteration Number					
Grid	Cold Flow	Fast Reaction	Finite Rate	Convergence	CPU Time	Machine
49x21x21 (21,609)	10	30	120	5.17 orders	1.77h	Cray-YMP
53x29x29 (44,573)	10	30	120	3.61 orders	3.57h	Cray-YMP
49x65x65 (207,025)	20	30	200	3.30 orders	21.3h	Cray-YMP

The test case shown here is strong swirling combustion in a 3-D gas turbine combustor. Figure 2 shows the inlet velocity vectors; the fuel and air enter the combustor coaxially with strong circulation. Figure 3 shows the calculated temperature isotherms on the center plane. The velocity vectors are plotted in Figure 4. The distributions of main chemical species CH_4, O_2, CO_2, H_2O and CO are presented in form of isopleths in Figures 5-9. A total of 160 time steps are used for this computation, including 10 steps for cold flow, 30 steps for fast reaction and 120 steps for the detailed finite-rate reaction. During each iteration step, 2 V-multigrid-cycles are performed for the flow part and 2 iterations for combustion part. For a 49x65x65 grid of 207,025 grid points, the calculation takes only about 200 time steps for finite rate calculation and 21.3 CRAY-YMP hours to reduce residuals by three orders of magnitude for all governing equations, demonstrating the high efficiency and capability of the present method.

Acknowledgement

The authors are grateful to Dr. John Goodrich at NASA Lewis Research Center for his helpful discussion. The authors also thank NASA Lewis Research Center for the sponsorship of this project.

REFERENCES

[1] Smooke, M.D., Mitchell, R.E., and Keys, D.E., 'Numerical Solution of Two Dimensional Axisymmetric Laminar Diffusion Flames,' *Combustion Sci. and Tech,* Vol. 67, pp.85-122, 1989.

[2] Smooke, M., 'Numerical Modeling of Laminar Diffusion Flames,' *Numerical Approaches to combustion modeling*, Chapter 7, Edited by Oran and Boris, AIAA, pp.183-223, 1991.

[3] Peters, N., 'Length Scale in Laminar and Turbulent Flames,' *Numerical Approaches to combustion modeling,* Chapter 6, edit by E.S. Oran and J.P. Boris, AIAA, pp.155-182, 1991.

[4] Kailasanath, K., Laminar Flames in Premixed Gases, *Numerical Approaches to Combustion modeling,* Chapter 6, edit by E.S. Oran and J.P. Boris, AIAA, pp.225-252, 1991.

[5] Xu, Y., Smooke, M., Liu, P., and Long, M., 'Primitive Variable Modeling of Multidimensional Laminar Flames,' *Combust. Sci. and Tech.,* Vol. 90, pp.289-313, 1993.

[6] Shuen, J.S., 'Upwind Differencing and LU Factorization for Chemical Non-equilibrium Navier-Stokes Equations,' *J. of Computational Physics*, Vol. 99, No. 2, pp.233-250, 1992.

[7] Liao, C., Liu, Z. and Liu, C., 'Implicit Multigrid Method for Modeling 3-D Turbulent Diffusion Flames with Detailed Chemistry,' AIAA 95-0801, 33rd Aerospace Sciences Meeting and Exhibit, January, 1995, Reno, NV.

[8] Liu, Z., Liao, C., Liu, C. and McCormick, S., 'Multigrid Method for Multi-Step Finite Rate Combustion,' AIAA 95-0205, 33rd Aerospace Sciences Meeting and Exhibit, January, 1995, Reno, NV.

[9] Liu, C., Liu, Z., and McCormick, S., 'Multigrid methods for numerical simulation of laminar diffusion flames,' AIAA paper 93-0236, 31st Aerospace Sciences Meeting and Exhibit, January, 1993.

[10] Van Leer, B., 'Towards the Ultimate Conservative Difference Scheme V: A second-order Sequel to Gudonov's method,' *Journal of Computational Physics*, Vol. 32, pp.101-136, 1979.

[11] Anderson, W.K., Thomas, J.L. and Van Leer, B., 'Comparison of Finite Volume Flux Vector Splittings for the Euler Equations,' *AIAA Journal,* Vol. 24, No. 9, pp.1453-1460, 1986.

[12] Bai, X.S., and Fuchs, L., 'Calculations of Turbulent Combustion of Profane in Furnaces,' *International Journal for Numerical Methods in Fluids,* Vol. 17, pp.221-239, 1993.

[13] Kee, R.J., Rupley, F.M., and Miller, J.A., 'CHEMKIN-II: A Fortran Chemical Kinetics Package for the Analysis of Gas Phase Chemical Kinetics,' Sandia National Laboratories Report, SAND89-80093, UC-706, 1989.

[14] Koren, B., Upwind Schemes, 'Multigrid and Defect Correction for the Steady Navier-Stokes Equations, '*Proceedings of the 11th International Conference on the Numerical Methods in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. G. Voigt, Springer-Verlag, Berlin, 1989.

[15] Liu,C. and Liu,Z., 'High order difference and multigrid methods for spatially-evolving instability in a planar channel,' *J. Comput. Phys.,* Vol. 106, pp.92-100, 1993.

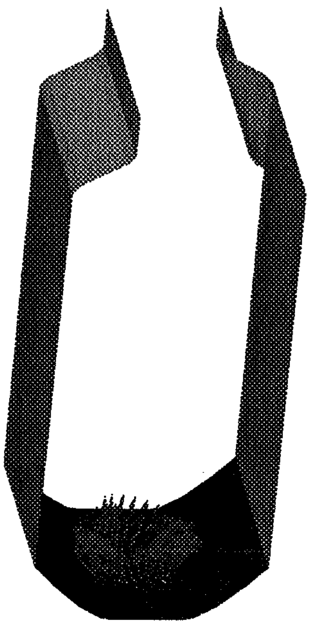


Figure 2: Velocity vectors at the inlet

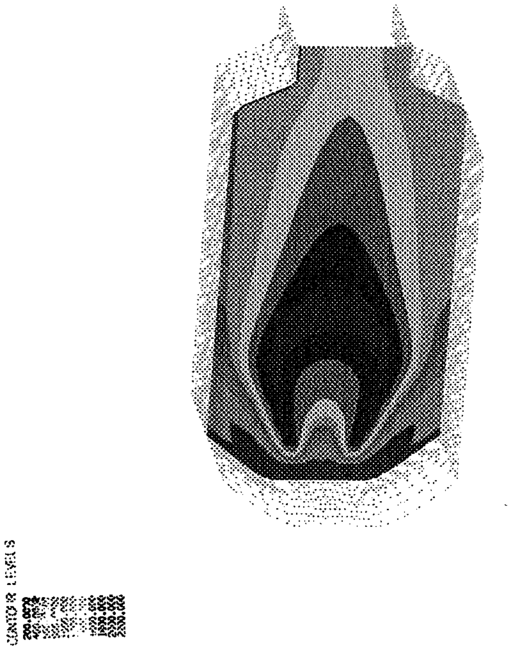


Figure 3: Temperature isotherms on the center (x, y)-plane

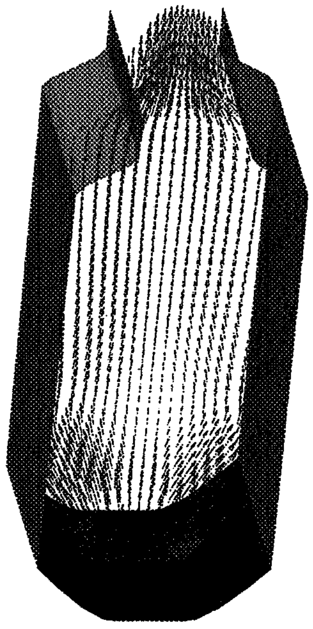


Figure 4: Vector plots of the flow field on the center (x, y)-plane (laminar)

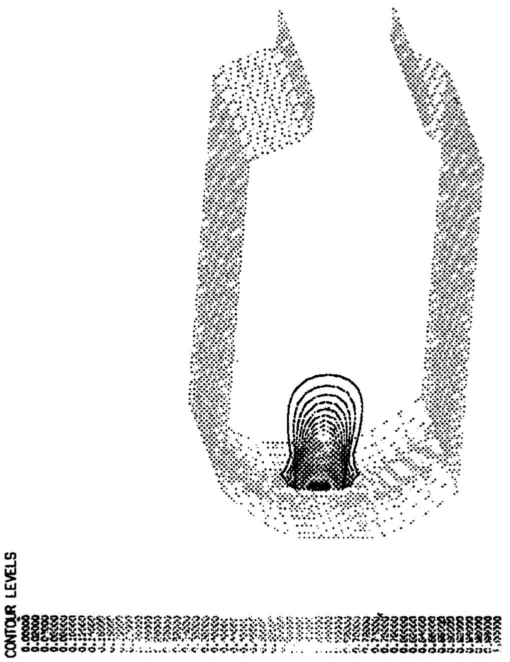


Figure 5: CH_4 isopleths (mass fraction) on the center (x, y)-plane

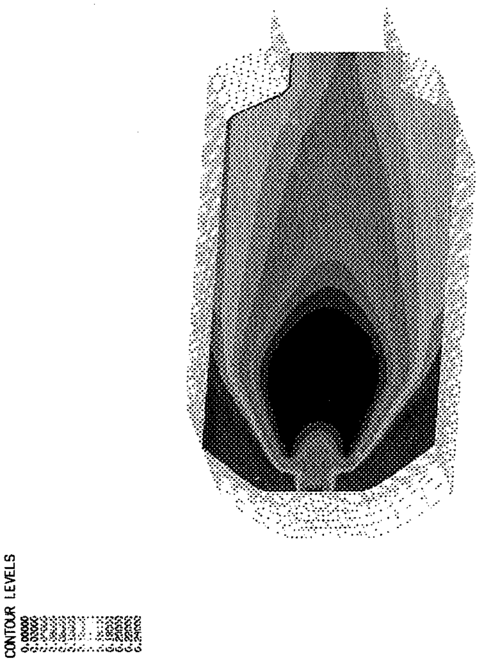


Figure 6: O_2 isopleths (mass fraction) on the center (x, y)-plane

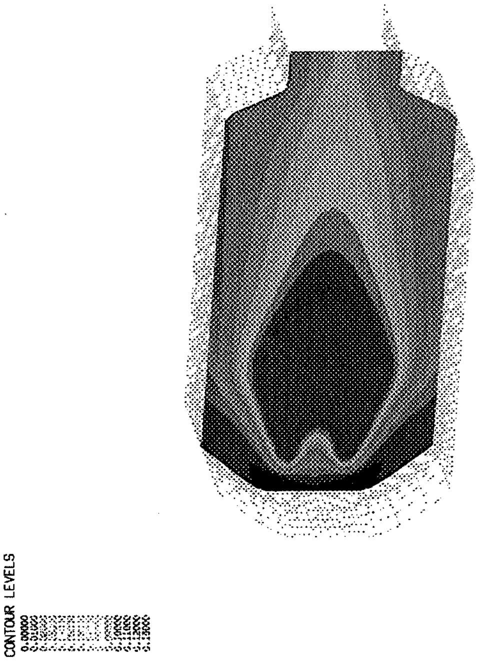


Figure 8: H_2O isopleths (mass fraction) on the center (x, y)-plane

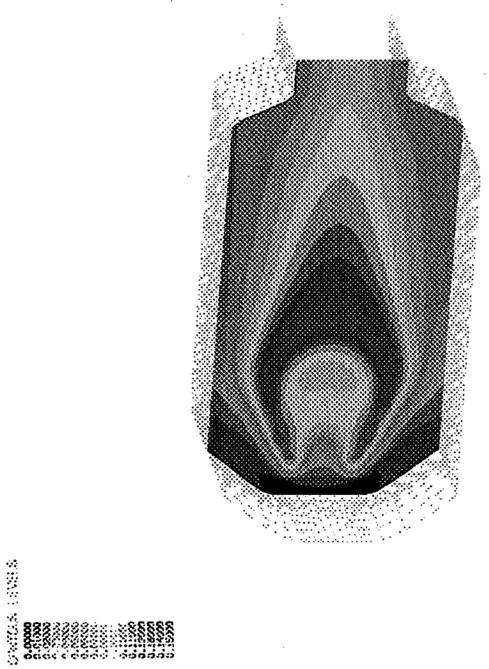


Figure 7: CO_2 isopleths (mass fraction) on the center (x, y)-plane

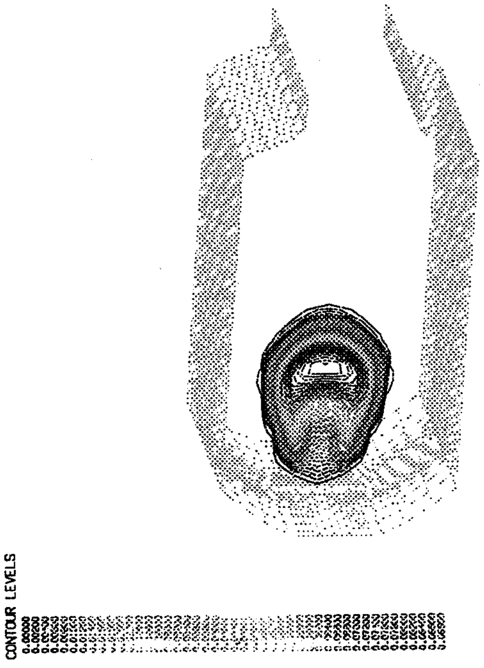


Figure 9: CO isopleths (mass fraction) on the center (x, y)-plane

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Conference Publication

4. TITLE AND SUBTITLE
Seventh Copper Mountain Conference on Multigrid Methods

5. FUNDING NUMBERS
WU 505-59-53-01

6. AUTHOR(S)
N. Duane Melson, Tom A. Manteuffel, Steve F. McCormick,
and Craig C. Douglas, Editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17593B

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

Department of Energy
Washington, DC 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CP-3339
Part 2

11. SUPPLEMENTARY NOTES
Organizing Institutions: University of Colorado at Denver; Front Range Scientific Computations, Inc.; and the Society for Industrial and Applied Mathematics

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 64
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Seventh Copper Mountain Conference on Multigrid Methods was held on April 2–7, 1995 at Copper Mountain, Colorado. This book is a collection of many of the papers presented at the conference and so represents the conference proceedings. NASA Langley graciously provided printing of this document so that all of the papers could be presented in a single forum. Each paper was reviewed by a member of the conference organizing committee under the coordination of the editors.

The multigrid discipline continues to expand and mature, as is evident from these proceedings. The vibrancy in this field is amply expressed in these important papers, and the collection shows its rapid trend to further diversity and depth.

14. SUBJECT TERMS
Multigrid; Algorithms; Computational fluid dynamics (CFD)

15. NUMBER OF PAGES
431

16. PRICE CODE
A19

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT