Electronic Transactions on Numerical Analysis.
Volume 10, 2000, pp. 74-91.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A HYBRID MULTIGRID METHOD FOR THE STEADY-STATE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS *

MICHAEL PERNICE†

Abstract. Multigrid methods for solving the steady-state incompressible Navier-Stokes equations require an appropriate smoother and coarse grid solution strategy to be effective. Classical pressure-correction methods, such as SIMPLE and SIMPLER, are widely used as solvers in engineering analysis codes, but can also be used as effective multigrid smoothers. An inexact Newton method preconditioned by a linear multigrid method with a pressure-correction smoother can serve as a coarse grid solver. A hybrid nonlinear multigrid scheme based on combinations of these components is described. A standard benchmark problem is used to demonstrate the effectiveness of SIMPLER smoothing and the impact an inexact Newton coarse grid solver has on the resulting nonlinear multigrid scheme.

Key words. multigrid, Newton-Krylov methods, pressure-correction methods, incompressible fluid dynamics.

AMS subject classifications. 65N55, 65H10.

1. Introduction. Efficient solution of the steady-state incompressible Navier-Stokes equations is of considerable interest in computational science and engineering. Strategies for solving these equations include fixed point iterative methods such as SIMPLE [26] and SIMPLER [25], multigrid methods [3], and inexact Newton methods [12, 6, 13]. The fixed point methods generally have low storage overhead and are straightforward to implement, but suffer from slow rates of convergence. The full approximation storage (FAS) method combines somewhat higher storage overhead with inexpensive iterations and high rates of convergence, but this is difficult to achieve without a suitable smoother and coarse grid solver. Newton's method provides superlinear rates of convergence, which are highly attractive, but Newton-based iterations are considerably more expensive than multigrid iterations, both in terms of operations and storage. The performance of Newton's method is sensitive to the choice of an initial approximation, even when globalization strategies are used. Recent advances in the development of inexact Newton methods have made this approach more competitive, but a good preconditioner is still necessary to achieve satisfactory performance.

Instead of considering these to be three competing and mutually exclusive approaches to solving the steady-state incompressible Navier-Stokes equations, it is beneficial to regard them as different components with complementary strengths and weaknesses that can be assembled in a number of ways to construct an effective solver. Pressure-correction methods can be used as smoothers in either linear or nonlinear multigrid methods. The expense of an inexact Newton method can be mitigated by using it as the coarse grid solver for FAS, which will often supply a good initial approximation to the Newton method. An improved coarse grid solver can also improve the robustness of a nonlinear multigrid scheme, offsetting the potentially adverse effects of an inappropriately discretized coarse grid problem. As demonstrated in [27], pressure-correction methods and linear multigrid methods with a pressure-correction smoother can also be used to precondition the inexact Newton coarse grid solver.

Hybrid methods that consist of suitable combinations of these approaches have the potential to leverage existing engineering analysis packages, improving their performance and

*Received May 25, 1999. Accepted for publication February 1, 2000. Recommended by J. Dendy. Work supported by Reaction Engineering International, Salt Lake City, Utah through funding from the Department of Energy Office of Energy Research under grant DE–FG03–96ER82268; the University of Utah Center for the Simulation of Accidental Fires & Explosions, funded by the Department of Energy, Lawrence Livermore National Laboratory, under subcontract B341493; and the University of Utah Center for High Performance Computing.
†Center for High Performance Computing, University of Utah, Salt Lake City, Utah 84112-9050.

expanding the scope of problems that can be solved. In doing so, patterns of use of the older methods may need to be re-examined in order to realize the benefits of a hybrid approach. Algorithmic components that can be combined in a variety of ways to construct hybrid solvers are described in §2. The effectiveness of these combinations is examined in §3. These results are summarized and some conclusions are drawn in §4.

2. Algorithms. Before describing the separate algorithmic components that can be composed into a hybrid scheme, some notation and terminology is first established. The incompressible Navier-Stokes equations in conservative form may be written

$$\begin{aligned} (uu)_x + (uv)_y - \tfrac{1}{Re}\Delta u + p_x &= f_1 \\ (uv)_x + (vv)_y - \tfrac{1}{Re}\Delta v + p_y &= f_2 \\ u_x + v_y &= 0. \end{aligned} \tag{2.1}$$

A second-order centered discretization of these equations on a staggered grid produces a set of nonlinear equations that can be written in block matrix form as

$$\begin{pmatrix} Q_1 & 0 & \mathcal{G}_x^h \\ 0 & Q_2 & \mathcal{G}_y^h \\ \mathcal{D}_x^h & \mathcal{D}_y^h & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ 0 \end{pmatrix}$$

where $Q_i = Q_i(u,v)$. In this, $\mathcal{G}_x^h$, $\mathcal{G}_y^h$ are discrete gradient operators defined on a staggered grid with mesh size h in the respective directions, and $\mathcal{D}_x^h$, $\mathcal{D}_y^h$ are discrete divergence operators defined on a staggered grid with mesh size h in the respective directions. Because of the staggered relationship among variables, it follows that

$$\mathcal{G}_x^h = {\mathcal{D}_x^h}^T, \qquad \mathcal{G}_y^h = {\mathcal{D}_y^h}^T.$$

The discrete momentum transport operator is denoted

$$\mathbf{Q} = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}, \tag{2.2}$$

the discrete gradient operator is given by

$$\nabla^h = \begin{pmatrix} \mathcal{G}_x^h \\ \mathcal{G}_y^h \end{pmatrix}$$

and the discrete divergence operator is

$$\nabla^h \cdot = \begin{pmatrix} \mathcal{D}_x^h & \mathcal{D}_y^h \end{pmatrix}.$$

2.1. Multigrid Methods. The nonlinear multigrid scheme considered here is the standard Full Approximation Storage (FAS) method [3]. Following is a recursive version of a V-cycle that is appropriate for nonlinear problems. In this, I_h^{2h} is a restriction operator, I_{2h}^h is a prolongation operator, and B is a smoothing operator. Alternative grid cycling strategies, such as W-cycles or full multigrid, are not considered.

Algorithm FAS: Full Approximation Storage

PROCEDURE FAS-V(h, L^h, x^h, f^h)

IF $h = h_c$ THEN:

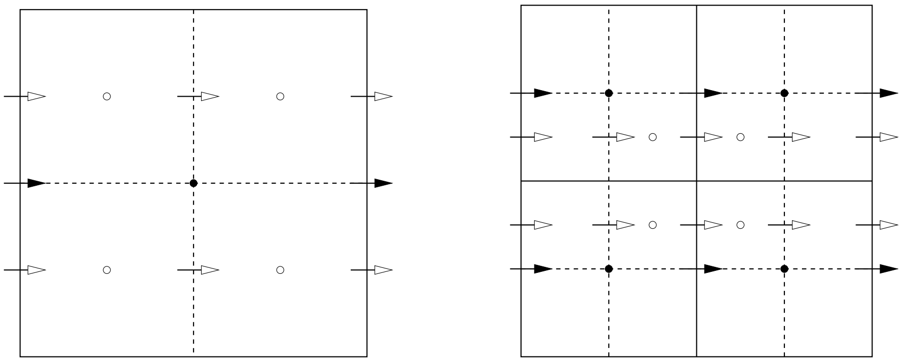

FIG. 2.1. *Relationships on a staggered grid between coarse (filled symbols) and fine values. Coarse cells are outlined with solid lines while fine cells use dashed lines. Relationships for restricting data to the coarse grid are depicted on the left, while relationships for prolonging data to the fine grid are depicted on the right. Only relevant fine grid data is shown; similar relationships hold for data centered on horizontal faces.*

SOLVE $L^h(x^h) = f^h$.
ELSE
PRESMOOTH $x^h \longleftarrow x^h + B(f^h - L^h(x^h))\ \nu_1$ TIMES.
SET $x^{2h} = I_h^{2h} x^h$.
RESTRICT $f^{2h} = I_h^{2h}(f^h - L^h(x^h)) + L^{2h}(x^{2h})$.
FAS-V($2h$, L^{2h}, x^{2h}, f^{2h}).
CORRECT $x^h = x^h + I_{2h}^h(x^{2h} - I_h^{2h} x^h)$.
POSTSMOOTH $x^h \longleftarrow x^h + B(f^h - L^h(x^h))\ \nu_2$ TIMES.

Most components of the FAS scheme considered here are more or less standard. Grid coarsening is readily achieved by defining each coarse cell to be a union of underlying fine cells. This geometry and the geometry of the staggered grid variable arrangement provide guidelines for constructing appropriate restriction and prolongation operators. These relationships are illustrated in Figure 2.1. Coarse cell-centered variables are the averages of the corresponding fine cell variables. Coarse face-centered variables are obtained by averaging fine values that reside on same cell face. Fine cell-centered variables are interpolated linearly from their four surrounding coarse cell neighbors. Prolongation of face-centered variables is done in two steps. First, adjacent coarse values are averaged to obtain fine values centered on the same face. These fine values are then averaged to obtain fine values on faces that lie between coarse faces.

The choices of restriction for the velocity and pressure have the desirable consequence that mass is conserved under coarsening:

$$\mathcal{D}_x^{2h}(I_h^{2h} u) + \mathcal{D}_y^{2h}(I_h^{2h} v) = I_h^{2h}(\mathcal{D}_x^h u + \mathcal{D}_y^h v).$$

In this, the restriction operators I_h^{2h} each must be interpreted in terms of the centering used for the variable that is operated on. Referring to Figure 2.1, this is simply a statement that the mass flux in the coarse cell is the same as the net mass flux of the four fine cells. As a consequence of these choices of restriction in the FAS algorithm, all solutions computed on the coarser grid levels are required to conserve mass.

Several choices for smoothing operations are possible. Brandt [4] advocates the use of *distributed Gauss-Seidel* smoothing. Vanka [37] employs a symmetric coupled Gauss-Seidel scheme in which mass conservation is enforced locally on a cell-by-cell basis. Wittum [39] introduces the notion of *transforming smoothers* and combines this idea with incomplete factorization to develop robust smoothing strategies. Shaw and Sivaloganathan [33, 34] demonstrated that classic single-grid solvers such as SIMPLE can be effective multigrid smoothers. Since much of this work was motivated by the desire to leverage existing SIMPLE-based single grid applications, the use of pressure-correction algorithms, in particular SIMPLER, as Navier-Stokes multigrid smoothers was investigated. These methods are described in §2.2.

The final component of a multigrid method is the selection of a coarse grid solver. There are very few circumstances in which a grid can be coarsened to the extent that it is practical to solve the coarse grid problem exactly, through direct or iterative means. This is due to several factors. First, grid geometries encountered in practice can be highly irregular and can only be coarsened to a modest degree. Too much coarsening can fail to resolve geometric features that affect the solution, rendering coarse grid corrections almost useless. New strategies based on agglomeration [21, 29, 22, 9] and Cartesian cell discretization methods [17] are promising, but these approaches can still produce coarse grid problems that are themselves of considerable size. Grid coarsening can also reveal weaknesses of certain discretization strategies, which can introduce too much numerical diffusion into the coarse grid problem and render the coarse grid correction almost useless [5]. Newton's method can be used as a coarse grid solver in a nonlinear multigrid scheme; an inexact variant of Newton's method that is used for this purpose is described in §2.3.

2.2. Pressure-Correction Methods. The SIMPLE family of algorithms is a widely-used set of techniques for incompressible fluid calculations. The descriptions that follow are restricted to two dimensional flows; their generalization to three dimensional problems is straightforward.

2.2.1. SIMPLE. The SIMPLE algorithm, introduced in [26], begins by approximately solving the discrete momentum equations, and then uses the discretized form of the mass conservation equation to derive an equation whose solution is used to update the pressure field and to correct the velocity field so that mass is conserved. It is possible to represent this process as a stationary iterative method [38]. To begin with, the discrete momentum transport operator $\mathbf{Q}$ is updated to reflect the current approximate solution $\mathbf{u}^{(k)}$ to the momentum equations. The resulting linearized momentum equations can then be solved to determine an intermediate velocity field $\mathbf{u}^{(k+\frac{1}{2})}$ using the current approximate pressure field $p^{(k)}$:

$$\mathbf{Q}\mathbf{u}^{(k+\frac{1}{2})} = \mathbf{f} - \nabla^h p^{(k)}. \tag{2.3}$$

Here

$$\mathbf{u}^{(k)} = \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}.$$

The solution of (2.3) does not have to be accurate and is usually accomplished with a few sweeps of a stationary iterative method, such as point or line Gauss-Seidel. The resulting velocity field $\mathbf{u}^{(k+\frac{1}{2})}$ does not conserve mass, and $\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})}$ is used to compute a correction δp to the pressure field whose gradient is also used to correct $\mathbf{u}^{(k+\frac{1}{2})}$.

To derive an equation for δp, let

$$\mathbf{D} = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}, \tag{2.4}$$

where $D_i = \mathrm{diag}(Q_i)$, and introduce the approximations

$$\begin{array}{rclcl} \mathbf{u}^{(k+\frac{1}{2})} & \approx & \mathbf{D}^{-1}\mathbf{Q}\mathbf{u}^{(k+\frac{1}{2})} & = & \mathbf{D}^{-1}(\mathbf{f} - \nabla^h p^{(k)}) \\ \mathbf{u}^{(k+1)} & \approx & \mathbf{D}^{-1}\mathbf{Q}\mathbf{u}^{(k+1)} & = & \mathbf{D}^{-1}(\mathbf{f} - \nabla^h p^{(k+1)}). \end{array} \tag{2.5}$$

ETNA
Kent State University
etna@mcs.kent.edu

Subtracting these equations and changing the approximation to equality leads to

$$\delta\mathbf{u} \equiv \mathbf{u}^{(k+1)} - \mathbf{u}^{(k+\frac{1}{2})} = -\mathbf{D}^{-1}\nabla^h \delta p, \tag{2.6}$$

where $\delta p = p^{(k+1)} - p^{(k)}$. This shows that $\mathbf{u}^{(k+\frac{1}{2})}$ can be corrected using a scaled gradient of the correction to the pressure field. Applying the discrete divergence operator to these equations gives

$$\nabla^h \cdot (\mathbf{u}^{(k+1)} - \mathbf{u}^{(k+\frac{1}{2})}) = \nabla^h \cdot \mathbf{D}^{-1}\nabla^h \delta p.$$

Finally, requiring that $\nabla^h \cdot \mathbf{u}^{(k+1)} = 0$ leads to

$$S\delta p = \nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})}, \tag{2.7}$$

where

$$S = -\nabla^h \cdot \mathbf{D}^{-1}\nabla^h = -(\mathcal{D}_x^h D_1^{-1}\mathcal{G}_x^h + \mathcal{D}_y^h D_2^{-1}\mathcal{G}_y^h) \tag{2.8}$$

is symmetric. Equation (2.7) is a generalized Poisson equation which must be solved for the pressure correction δp. Since the intended correction (2.6) should be 0 at locations where the velocity field is specified, (2.7) is supplemented with homogeneous boundary conditions at these locations. In particular, for problems where the velocity field is specified at the boundaries, δp is determined only up to an additive constant, and S is positive semi-definite. Summarizing:

Algorithm SIMPLE:

DETERMINE $\mathbf{u}^{(k+\frac{1}{2})}$ BY SOLVING (2.3).
FIND THE PRESSURE CORRECTION δp FROM (2.7).
CALCULATE THE VELOCITY CORRECTIONS $\delta\mathbf{u}$ USING (2.6).
UPDATE THE PRESSURE

$$p^{(k+1)} = p^{(k)} + \delta p$$

AND THE VELOCITIES

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k+\frac{1}{2})} + \delta\mathbf{u}.$$

Practical implementations usually employ underrelaxation, both in solving (2.3) and in applying the corrections δp and $\delta\mathbf{u}$. In this work, an underrelaxtion factor of 0.6 is used, together with 5 sweeps of point Gauss-Seidel for the momentum equations and 20 sweeps of point Gauss-Seidel for the pressure equation. SIMPLEC is a variation that replaces the entries in the diagonal matrices D_i with absolute rowsums from Q_i [36]; it is this variation that is actually used in the evaluations presented in §3.1.

2.2.2. SIMPLER. SIMPLER is a variation of SIMPLE due to Patankar [25]. It is similar to SIMPLE, but it determines $p^{(k+1)}$ from $\mathbf{u}^{(k)}$ and uses a separate potential field ϕ to enforce mass conservation in a manner similar to projection methods [10, 1].

As with SIMPLE, each cycle begins with an update of the momentum transport operator (2.2) to reflect the latest approximate solution $\mathbf{u}^{(k)}$. For the next iteration the pressure and velocity field should satisfy

$$\mathbf{Q}\mathbf{u}^{(k+1)} = \mathbf{f} - \nabla^h p^{(k+1)}.$$

To determine an equation for $p^{(k+1)}$, introduce the splitting $\mathbf{Q} = \mathbf{D} - (\mathbf{L} + \mathbf{U})$, where $\mathbf{D}$ is again given by (2.4) and $-(\mathbf{L} + \mathbf{U})$ contains the off-diagonal elements of $\mathbf{Q}$. It follows that

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \mathbf{D}^{-1}(\mathbf{f} + (\mathbf{L} + \mathbf{U})\mathbf{u}^{(k+1)} - \nabla^h p^{(k+1)}) \\ &\approx \mathbf{D}^{-1}(\mathbf{f} + (\mathbf{L} + \mathbf{U})\mathbf{u}^{(k)} - \nabla^h p^{(k+1)}). \end{aligned} \tag{2.9}$$

Taking the divergence of both sides of (2.9), requiring that $\nabla^h \cdot \mathbf{u}^{(k+1)} = 0$, changing the approximation to equality, and rearranging terms leads to

$$Sp^{(k+1)} = -\nabla^h \cdot \mathbf{D}^{-1}(\mathbf{f} + (\mathbf{L} + \mathbf{U})\mathbf{u}^{(k)}), \tag{2.10}$$

where S is again given by (2.8).

Once $p^{(k+1)}$ is known, the velocity field is updated by first solving

$$\mathbf{Q}\mathbf{u}^{(k+\frac{1}{2})} = \mathbf{f} - \nabla^h p^{(k+1)} \tag{2.11}$$

and then correcting $\mathbf{u}^{(k+\frac{1}{2})}$. This is done using the gradient of an auxiliary variable ϕ. To determine an equation for ϕ, the following *ansatz* is made for the correction:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k+\frac{1}{2})} - \mathbf{D}^{-1}\nabla^h \phi. \tag{2.12}$$

An equation for ϕ is then determined by taking the divergence of (2.12) and requiring that $\nabla^h \cdot \mathbf{u}^{(k+1)} = 0$:

$$S\phi = -\nabla \cdot \mathbf{u}^{(k+\frac{1}{2})}, \tag{2.13}$$

where S is again given by (2.8). Summarizing:

Algorithm SIMPLER:

DETERMINE $p^{(k+1)}$ BY SOLVING (2.10).
DETERMINE $\mathbf{u}^{(k+\frac{1}{2})}$ BY SOLVING (2.11).
DETERMINE ϕ BY SOLVING (2.13).
CORRECT $\mathbf{u}^{(k+\frac{1}{2})}$ USING (2.12).

The improved approximation (2.9) reduces the need for underrelaxation, though it is still used in practice. The momentum equations are solved with 4 sweeps of point Gauss-Seidel relaxation, and the pressure equation (2.10) is solved with 20 sweeps. Instead of solving (2.10) directly, a residual equation is solved for δp, following [36], and an underrelaxation factor of 0.8 is used to complete the pressure update.

Note that the correction (2.12) may be rewritten as follows:

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \mathbf{u}^{(k+\frac{1}{2})} + \mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})} \\ &= \left(I + \mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h \cdot\right) \mathbf{u}^{(k+\frac{1}{2})} \\ &\equiv \mathcal{P}\mathbf{u}^{(k+\frac{1}{2})}. \end{aligned}$$

From (2.8) it follows that $\mathcal{P}^2 = \mathcal{P}$. Thus $\mathcal{P}$ is actually a *projection* (though it is not an *orthogonal* projection with respect to the standard inner product).

An interesting feature of SIMPLER is the use of ϕ to enforce mass conservation. Note that the derivation of (2.13) is unconventional. Patankar [25] derives the *ansatz* (2.12) through a process similar to the one that led to (2.6). Because of this approximation, the conventional wisdom is that ϕ need not be calculated with high accuracy, and usually only a few sweeps of a stationary iterative method are applied. However the derivation based on (2.12) shows clearly that the residual in mass conservation is directly related to the accuracy with which (2.13) is solved:

$$\|\nabla^h \cdot \mathbf{u}^{(k+1)}\| = \|\nabla^h \cdot \mathcal{P}\mathbf{u}^{(k+\frac{1}{2})}\| = \|\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})} + S\phi\|.$$

Thus, the conventional approach may fail to produce a mass-conserving velocity field, since just a few sweeps of a stationary iterative method may not reduce the residual of (2.13) by very much. A guaranteed improvement can be achieved by solving (2.13) to a prescribed accuracy:

$$\|\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})} + S\phi\| \leq tol \tag{2.14}$$

where tol can be either an absolute or a relative accuracy. Doing so increases the cost of each iteration of SIMPLER, and the additional cost must be offset by an improved rate of convergence for the overall scheme. Thus, a very efficient solver should be used to enforce (2.14). Since S is symmetric and positive semi-definite, a preconditioned conjugate gradient method can be used. The efficiency of this approach depends on using an effective preconditioner. This work employs a multigrid preconditioner, which is composed of volume averaged restrictions, piecewise constant prolongation, symmetric Gauss-Seidel smoothing, V(1,1) cycles, and a Galerkin coarse grid version of (2.8). This latter choice was made necessary by the fact that $\mathbf{D}$ is not conveniently available on the coarser grids, and was made easy to implement because of the choices made for intergrid transfers.

It is also possible to enforce a condition similar to (2.14) in the solve for the pressure correction (2.7) in SIMPLE. However it was found that this was accompanied by a need to damp δp even further in order to achieve convergence. Doing so slowed convergence of the overall method, making it non-competetive. This effect was not observed with SIMPLER, where ϕ is used solely to correct $\mathbf{u}^{(k+\frac{1}{2})}$.

2.3. Inexact Newton Methods. Newton's method for solving a system of nonlinear equations

$$F(x) = 0, \qquad F : \mathbb{R}^n \to \mathbb{R}^n, \tag{2.15}$$

requires, at the kth step, the solution of the linear *Newton equation*

$$F'(x_k)\, s_k = -F(x_k), \tag{2.16}$$

where x_k is the current approximate solution and F' is the Jacobian matrix of the system. Once the Newton step s_k is determined the current approximation is updated via

$$x_{k+1} = x_k + s_k.$$

This process is continued until a satisfactory solution is found, which is usually judged by making $\|F(x_k)\|$ or $\|s_k\|$ (or both) sufficiently small. Traditionally, Newton's method was considered to be inappropriate for the solution of large-scale systems of nonlinear equations because of the high computational and storage costs of solving (2.16). However exact solution of (2.16) is not necessary for Newton's method to converge. In a *Newton iterative method*, or *truncated Newton method*, (2.16) is replaced by an *inexact Newton condition* [12]

$$\|F(x_k) + F'(x_k)\, s_k\| \leq \eta_k \|F(x_k)\|, \tag{2.17}$$

and so a suitable inexact Newton step s_k can be found with an iterative method at a significant reduction in cost. The "forcing term" $\eta_k \in [0, 1)$ in (2.17) can be specified in several ways; superlinear and even quadratic convergence of the inexact Newton method can be obtained under certain choices of the forcing terms [12, 14]. The best choice of η_k is somewhat problem-specific (see for example [32] for some comparative studies). In this work, the strategy labeled Choice 1 in [14] is used to select the forcing terms. Newton iterative methods are especially well-suited for large-scale problems and have been used very successfully in a number of applications [8, 18, 19, 32, 7, 2].

There are many ways to compute an inexact Newton step s_k, and the efficiency of an inexact Newton method is strongly affected by this choice. Krylov subspace methods [16] are especially well-suited for this purpose since they only require matrix-vector products $F'(x_k)v$. This further specialization of inexact Newton methods leads to the class of methods referred to as *Newton-Krylov methods*. The matrix-vector products needed in a Newton-Krylov method may be approximated with finite differences of function values

$$F'(x_k)v \approx \frac{F(x_k + \epsilon v) - F(x_k)}{\epsilon}, \tag{2.18}$$

and so the Jacobian F' never needs to be explicitly formed. This approach is frequently referred to as a *matrix-free Newton-Krylov* method. While this greatly reduces storage requirements and simplifies implementation, the differencing parameter ϵ must be chosen carefully to balance different sources of errors and differences in scales that may be present in v. Furthermore, some information about the Jacobian is still neeeded to construct a preconditioner. Finally, matrix-free Newton-Krylov methods generally require more nonlinear iterations than a Newton-Krylov method that uses the Jacobian directly.

Among Krylov subspace methods, GMRES [31] is generally preferred, since it minimizes the residual at every iteration. Unfortunately, in order to enforce this, storage requirements grow linearly and work requirements grow quadratically with the number of iterations. In practice this is dealt with by restarting the method, which can potentially slow down convergence or even cause divergence if restarting is done too frequently. While alternatives such as BiCGSTAB [35] and transpose-free QMR [15] can be used, they do not share the minimum residual property and are generally not as robust as GMRES, provided a sufficiently large restart value is used.

Effective preconditioning plays a critical role in the efficient computation of an inexact Newton step. For restarted GMRES, a good preconditioner not only accelerates convergence but also reduces storage requirements by allowing more frequent restarts. Incomplete LU (ILU) factorizations [24, 30] are popular choices, but they require information about the Jacobian that may be difficult to determine in a matrix-free inexact Newton method. Further, ILU factorizations can have high storage requirements if a lot of element fill-in is allowed. Calculating an incomplete factorization can also be computationally expensive, and this cost is multiplied by the number of times the preconditioner is updated during the nonlinear solution process. One approach to constructing an effective preconditioner is to use a linear multigrid method [20, 27]. When using the Newton-Krylov method as a coarse grid solver, this approach has the added advantage of reusing software components in the linear preconditioner that were used to implement the FAS smoother.

Finally, another traditional objection to using Newton's method for large-scale problems is the need to find a good initial approximation x_0. Newton's method (and its inexact counterpart) can fail to converge if x_0 is not chosen carefully. Fortunately, classical strategies for improving the likelihood of convergence from a poor initial approximation also apply to Newton iterative methods [13]. The backtracking globalization strategy given in Algorithm INB from [13] and implemented in NITSOL [28] is employed in this work.

2.4. Hybrid Methods. It is possible to assemble these algorithmic components into a number of different hybrid strategies. An FAS method can be constructed using either SIMPLE or SIMPLER as a smoother. These pressure-correction schemes can also be used as preconditioners for a Newton-Krylov method. Finally, the Newton-Krylov-MG method NK-MG [27] can be used as a coarse grid solver for FAS.

Using a pressure-correction method as a smoother for FAS is easy to accomplish. The current approximation u^{h_l} on grid level l is used to update $\mathbf{Q}^{h_l}$ and S^{h_l} through a rediscretization on the grid. The SIMPLE or the SIMPLER algorithm is then applied with source term $\mathbf{f}^{h_l}$ and initial approximation u^{h_l}, using the number of iterations specified by the number of pre- and post-smoothing steps. The result then is processed for use on other grids except at the end of the V-cycle, where it is the updated approximate solution.

Pressure-correction methods may be used as either preconditioners or linear multigrid smoothers in a manner analogous to that described above, with one slight change in interpretation. In performing either a preconditioning solve or a smoothing operation, a system of the form

$$Me = r$$

is solved using a preconditioner M, where r is a residual and e is a correction to the current approximation to the solution of the linear problem. When M is itself one or more iterations of an iterative method such as SIMPLE or SIMPLER, the initial approximation is $e = 0$, reflecting the expectation that only a relatively small correction will be calculated.

The hybrid approach of using a Newton-Krylov method as a coarse grid solver in an FAS procedure is accomplished by simply passing the problem

$$F(\mathbf{u}, p) = \begin{pmatrix} Q^{h_c}(\mathbf{u}) & \nabla^{h_c} \\ \nabla^{h_c}\cdot & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} - \begin{pmatrix} \mathbf{f}^{h_c} \\ 0 \end{pmatrix} = 0$$

to the Newton-Krylov solver, where h_c is the coarse grid, Q^{h_c}, f^{h_c} are the corresponding coarse grid operator and source term, and the initial approximation $u = u^{h_c}$ uses the current coarse grid solution. This strategy addresses several weaknesses. The Newton-Krylov method is usually supplied with a good initial approximation, reducing the need for backtracking and enabling early realization of superlinear convergence. The smaller problem size also helps to address the high storage cost of using GMRES within the Newton-Krylov method. In contrast to the level smoothing operations, the Newton-Krylov method produces a solution with a guaranteed amount of error reduction. Thus, the hybrid approach allows insertion of a method that is guaranteed to produce a solution with some desired accuracy at any level of a given grid hierarchy. This can help to stabilize and enhance the robustness of the overall FAS method. However this feature must be balanced with increased storage and computation costs that are incurred by introducing the Newton-Krylov method at finer levels of the grid hierarchy.

3. Numerical Evaluations. Hybrid approaches that combine multigrid methods, Newton-Krylov methods, and pressure-correction methods offer a wide range of algorithmic choices. Since these combinations embed iterative methods within iterative methods, it is difficult to predict the most effective combination. Numerical experimentation is necessary to help determine which hybrid approaches warrant further investigation. This section examines the performance of the nonlinear multigrid algorithm under various combinations of coarse grid solvers and smoothers.

Conventional evaluations of multigrid effectiveness usually employ some standardized measure of work unit per iteration. However, estimating work units is difficult when iterative methods are used to implement smoothers and/or coarse grid solvers. Consequently computational effort is measured relative to the CPU time needed to reduce the residual by a prescribed amount. CPU time on a MIPS R10000 processor needed to reach a relative reduction of the nonlinear residual by 10^{-6} is reported.[1]

Buoyancy-driven natural convection in an enclosed cavity is a standard benchmark problem that is frequently used to evaluate different numerical schemes and solution methods [11]. The governing equations consist of the incompressible Navier-Stokes equations (2.1) coupled to an energy transport equation

$$(uT)_x + (vT)_y - \frac{1}{RePr}\Delta T = 0$$

together with a body force on the fluid that, under the Boussinesq approximation, is proportional to the temperature

$$\mathbf{f} = \begin{pmatrix} 0 \\ \frac{Ra}{Re^2 Pr}T \end{pmatrix}.$$

In this, Re is the Reynolds number, Pr is the Prandtl number, and Ra is the Rayleigh number. Following [23], Re is fixed at 1, Pr is fixed at 0.71, and Ra is varied. The problem is defined on the unit square $\Omega = [0,1] \times [0,1]$ with boundary conditions

$$\begin{aligned} u = v = 0 &\quad \text{on } \partial\Omega \\ T(0,y) = 0,\ T(1,y) = 1 &\quad y \in [0,1] \\ T_y(x,0) = T_y(x,1) = 0 &\quad x \in [0,1]. \end{aligned}$$

In the staggered formulation, the temperature is treated as a cell-centered quantity. Except for those locations where face-centering places a velocity component on $\partial\Omega$, linearly extrapolated values are used to specify boundary conditions.

The additional transport equation is readily accommodated by the SIMPLE and SIMPLER smoothers. Five sweeps of the point Gauss-Seidel method are applied to the discretized transport equation prior to solving the momentum equations. The updated temperature field is then incorporated into **f** before solving the momentum equations.

3.1. Effectiveness of different smoothers. The effectiveness of SIMPLE and SIMPLER as FAS smoothers is evaluated in this section. To distinguish among smoother variants, FAS-SIMPLE refers to FAS with SIMPLE smoothing, and FAS-SIMPLER refers to FAS with SIMPLER smoothing. FAS-SIMPLE and FAS-SIMPLER were compared at several values of Ra. For $Ra \leq 1000$, FAS-SIMPLE was found to be $5 - 10\%$ faster than FAS-SIMPLER. More postsmoothing sweeps were generally needed to achieve this. However, for larger values of Ra, FAS-SIMPLE began to falter, and was outperformed by FAS-SIMPLER. A comparison of convergence histories at $Ra = 10,000$ appears in Figure 3.1. For lower values of Ra, the convergence history of FAS-SIMPLE is actually quite smooth, and the irregularity of the convergence history in this figure seemed to presage failure of the method for larger values of Ra. FAS-SIMPLE with V(2,1) cycles actually failed to converge for this case. Included in the figure is the convergence history for FAS-SIMPLER when (2.14) is enforced with $tol = 0.5\|\nabla \cdot \mathbf{u}^{(k+\frac{1}{2})}\|$ using the multigrid-preconditioned conjugate gradients method described at the end of §2.2.2. There is a clear, though modest, benefit.

[1]The results for the smaller problems were obtained on a processor running at 195 Mhz with a 1 MB L2 cache; for the larger problems a processor with a 4 MB L2 cache was used.

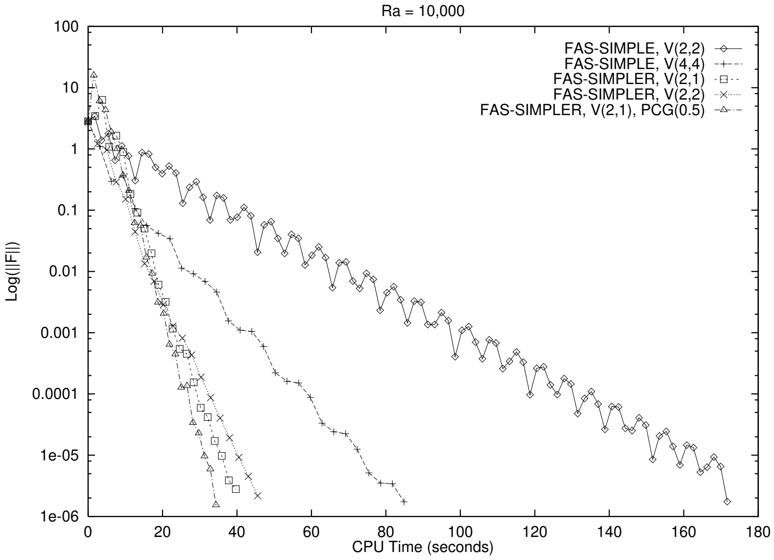

FIG. 3.1. *Convergence histories of FAS-SIMPLE and FAS-SIMPLER. Results obtained on a* 128×128 *grid with 3 grid coarsenings.*

3.2. Influence of coarse grid solution strategy. As was seen in the previous section, no substantial convergence difficulties were observed for $Ra \leq 10,000$. However, for $Ra > 10,000$, some instabilities were observed in the convergence histories. Here, solving the coarse grid problem with the Newton-Krylov method was found to be beneficial. For $Ra \gtrsim 30,000$, FAS-SIMPLER actually failed to converge unless the Newton-Krylov method was used for the coarse grid problem. For even larger values of Ra, it was found that enforcing (2.14) benefited *both* the smoother and the coarse grid solver. These results are examined in more detail below.

Since §3.1 demonstrated that FAS-SIMPLER is superior to FAS-SIMPLE, all results presented in this section were obtained using SIMPLER smoothing and the term FAS is no longer qualified by which smoother is employed. Names by which the various combinations of algorithms are referred to in this section are listed in Table 3.1. The NK-MG implementation always uses V(2,1) cycles with SIMPLE smoothing in the preconditioner, which was found in [27] to be the most efficient option.

The first set of results are for $Ra = 20,000$ and appear in Figure 3.2, which compares the effectiveness of varying numbers of pre- and post-smoothing sweeps. In this and in the remainder of this section, $\log(\|F\|_\infty)$ is plotted against CPU time. Here, V(2,1) cycles are the most efficient, and the addition of solving (2.14) with $tol = \delta\|\nabla \cdot \mathbf{u}^{(k+\frac{1}{2})}\|$ and smaller values of δ produced a minor additional improvement. Note that while decreasing δ reduces the number of iterations by 2 and smooths the convergence history somewhat, the additional cost per iteration slightly increases the overall computational cost. For this case, and for the cases discussed in the rest of this section, the multigrid-preconditioned conjugate gradients method described in §2.2.2 required only 2–3 iterations to satisfy (2.14).

The plot in Figure 3.3 depicts convergence histories obtained by using the Newton-Krylov method to solve the coarse grid problem to prescribed accuracy. For reference, the curves labeled “V(2,1), PCG(0.5)” is duplicated between Figures 3.2 and 3.3. Clearly, solving the coarse grid problem to prescribed accuracy is always beneficial, with the NK-MG coarse grid solver being the most efficient. Also note that the NK-MG coarse grid solver only

TABLE 3.1
Options evaluated for nonlinear multigrid components.

FAS	Multigrid with SIMPLER smoothing. Coarse grid solves use 16 sweeps of the smoother.
FAS/NK-SIMPLE(m)	Multigrid with SIMPLER smoothing and a Newton-Krylov coarse grid solver. The latter is preconditioned with m sweeps of SIMPLE.
FAS/NK-MG	Multigrid with SIMPLER smoothing and a Newton-Krylov coarse grid solver. The latter is preconditioned by a multigrid method that uses a V(2,1) cycle with SIMPLE smoothing.
FAS/PCG(δ)	Same as FAS, with (2.14) enforced with $tol = \delta\|\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})}\|$.
FAS/PCG(δ)/NK-SIMPLE(m)	Same as FAS/NK-SIMPLE(m), with (2.14) enforced with $tol = \delta\|\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})}\|$.
FAS/PCG(δ)/NK-MG	Same as FAS/NK-MG, with (2.14) enforced with $tol = \delta\|\nabla^h \cdot \mathbf{u}^{(k+\frac{1}{2})}\|$.

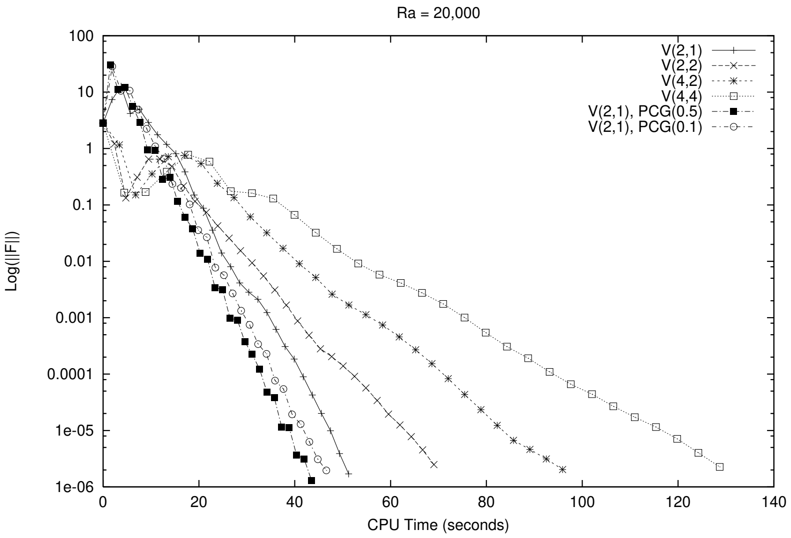

FIG. 3.2. *Convergence histories of* FAS *for $Ra = 20,000$. Different numbers of pre- and post-smoothing sweeps are compared. Results were obtained on a* 128×128 *grid with 3 grid coarsenings.*

required 2–3 iterations to reach a prescribed relative accuracy of 0.1; any smaller prescribed accuracy increased the cost of the cost grid solver and the overall cost of the calculation.

For $Ra = 50,000$, use of the Newton-Krylov coarse grid solver was essential, since both FAS and FAS/PCG(δ) failed to converge. Some typical results appear in Figures 3.4 and 3.5. Again for reference, the convergence histories labeled NK-SIMPLE(4) and NK-MG are the same in these two figures. The plot in Figure 3.4 shows that the NK-SIMPLE(4) coarse grid solver produces the best performance. However, the plot in Figure 3.5 shows that enforcing (2.14) enhances the performance of both FAS/NK-SIMPLE(4) and FAS/NK-MG, and that reducing tol further improves the performance of both.

ETNA
Kent State University
etna@mcs.kent.edu

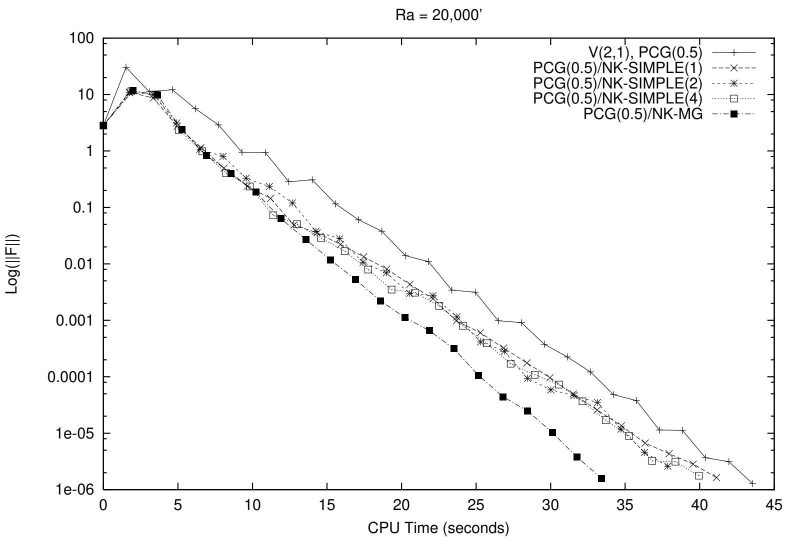

FIG. 3.3. *Convergence histories of* FAS *for* $Ra = 20,000$. *Different coarse grid solvers are compared. Results were obtained on a* 128×128 *grid with 3 grid coarsenings. The NK-MG option used an additional 3 grid coarsenings.*

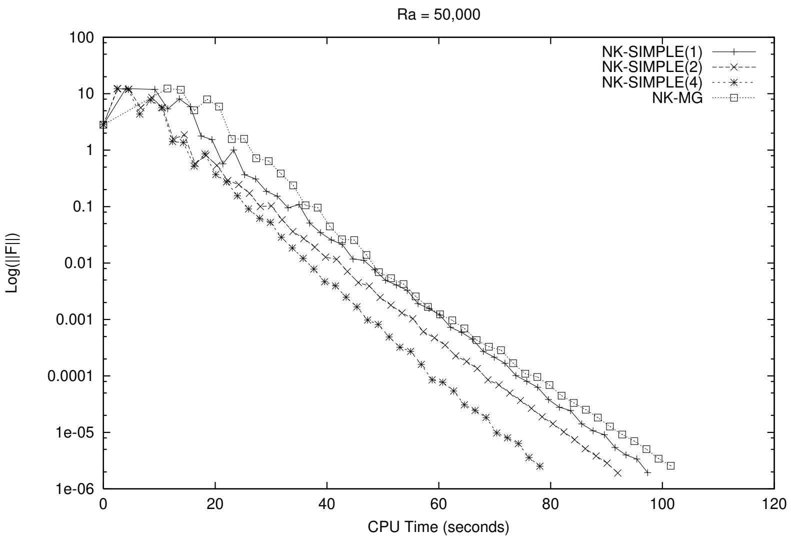

FIG. 3.4. *Convergence histories of* FAS *for* $Ra = 50,000$. *Different preconditioners in the coarse grid solver are compared. Results were obtained on* 128×128 *grid using 3 grid coarsenings.*

It is interesting to note the patterns of these improvements. FAS/PCG(0.5)/NK-SIMPLE(4) performs poorly on the second iteration, but this is subsequently overcome by a faster rate of convergence. FAS/PCG(0.1)/NK-SIMPLE(4) does not encounter this initial difficulty, and subsequently converges at about the same rate as FAS/PCG(0.5)/NK-SIMPLE(4). The strategies that employ the NK-MG coarse grid solver behave similarly, though initially FAS/PCG(0.5)/NK-MG does not encounter as much difficulty as FAS/PCG(0.5)/NK-SIMPLE(4). Both FAS/PCG(δ)/NK-

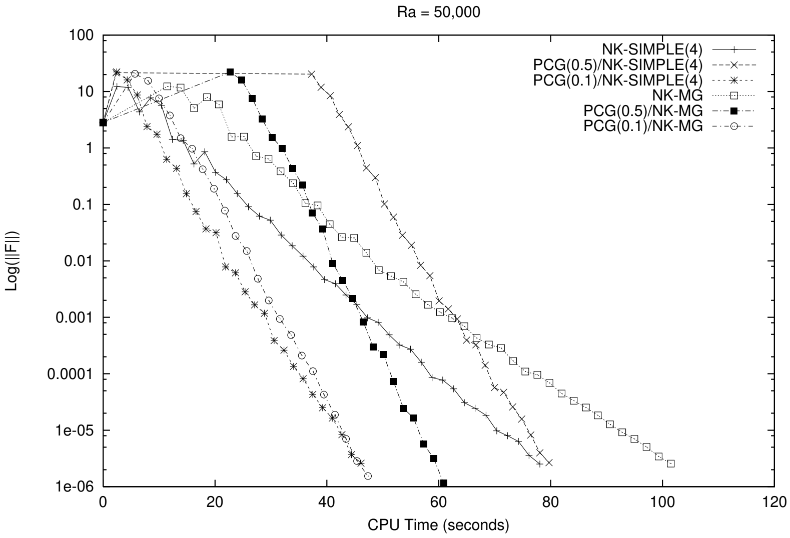

FIG. 3.5. *Convergence histories of* FAS *for* $Ra = 50,000$. *Different tolerances in the smoother are compared. Results were obtained on* 128×128 *grid using 3 grid coarsenings. The* NK-MG *coarse solver used 2 additional grid coarsenings.*

MG variations have higher rates of convergence than FAS/PCG(δ)/NK-SIMPLE(4).

The effect of enforcing (2.14) is even more dramatic at $Ra = 100,000$. While using the Newton-Krylov method is necessary to achieve convergence, it alone is not enough to guarantee efficiency. This is illustrated in Figures 3.6 and 3.7. Without enforcing (2.14), the method failed with V(2,1) cycles, so V(2,2) cycles were used to produce Figure 3.6. On the first few iterations, the Newton-Krylov coarse grid solver gets trapped in a backtracking loop, with attendant increase in the forcing term. The method eventually recovers, but at an enormous relative cost. Interestingly, the FAS/NK-SIMPLE(k) options, which use no further grid coarsenings, are not as sensitive to this phenomenon.

Enforcing (2.14) helps to address the difficulties experienced by the coarse grid solver as well as enabling convergence with V(2,1) cycles. This behavior is illustrated in Figure 3.7. Here the curve labeled NK-SIMPLE(4) is the same as the one that appears in Figure 3.6. The combination of using V(2,1) cycles and enforcing (2.14) improves the performance of the variant using the NK-SIMPLE(4) coarse grid solver by a factor of 3. While the variant using the NK-MG coarse grid solver is more expensive per iteration, FAS/PCG(0.5)/NK-MG requires one third fewer iterations than FAS/PCG(0.5)/NK-SIMPLE(4); this combination produces a net 18% additional improvement in performance.

It is interesting to note that the coarsest grid used for $Ra = 50,000$ and $Ra = 100,000$ were both 4×4. However for $Ra = 50,000$, the NK-MG coarse grid solver was applied on the 32×32 grid, while for $Ra = 100,000$ it was applied on the 64×64 grid. In the latter case the method failed when the NK-MG coarse grid solver was tried on the 32×32 grid. This illustrates the flexibility that is provided by the NK-MG coarse grid solver in deciding which is the coarsest grid in the FAS scheme, while still allowing the use of the entire grid hierarchy to efficiently resolve all components of the error.

Considerable difficulties were encountered when trying to apply these strategies to the case $Ra = 500,000$. No combination was found that enabled convergence with FAS/PCG(δ)/NK-SIMPLE(k) for $k = 4$. For $k = 2$, V(2,2) cycles failed to converge and V(4,4) cycles were

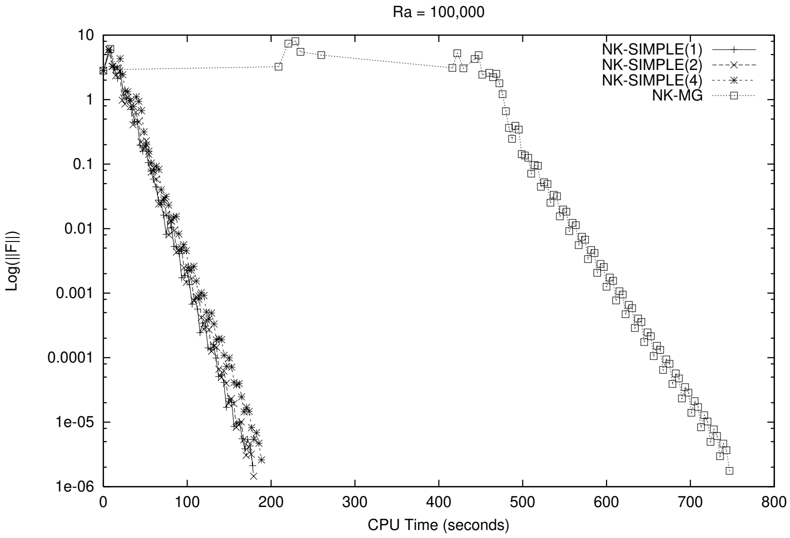

FIG. 3.6. *Convergence histories of* FAS *for $Ra = 100,000$. Different preconditioners in the coarse grid solver are compared. Results were obtained on a 128×128 grid with 2 grid coarsenings. The* NK-MG *coarse grid solver used an additional 3 grid coarsenings.*

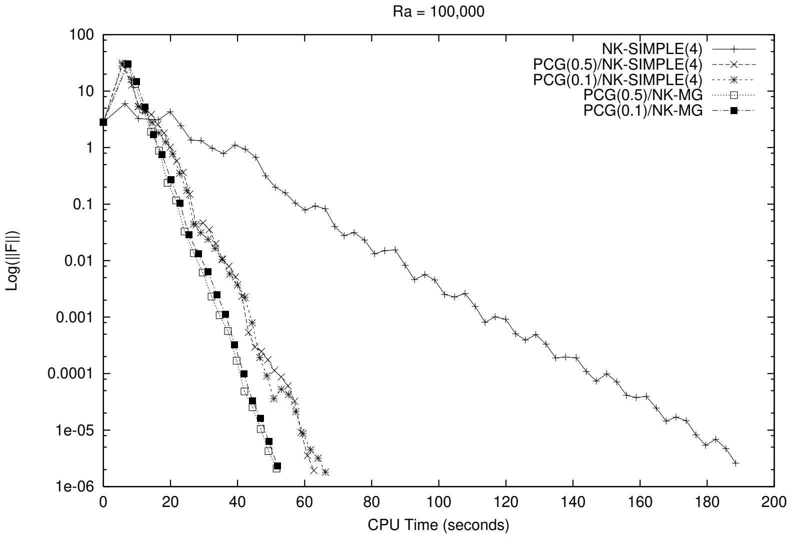

FIG. 3.7. *Convergence histories of* FAS *for $Ra = 100,000$. Different tolerances in the smoother are compared. Results were obtained on a 128×128 grid with 2 grid coarsenings. The* NK-MG *coarse grid solver used an additional 3 grid coarsenings.*

needed. Some of the remaining results are depicted in Figure 3.8. Convergence histories for FAS/PCG(δ)/NK-SIMPLE(1) with V(4,4) cycles are even more erratic than the one shown for V(2,2) cycles and are omitted. All FAS/PCG(δ)/NK-MG convergence histories are quite satisfactory, showing no signs of instability and rates of convergence on the order of $\rho \approx 0.6$.

Table 3.2 summarizes the results of this section in terms of the best convergence rates that

ETNA
Kent State University
etna@mcs.kent.edu

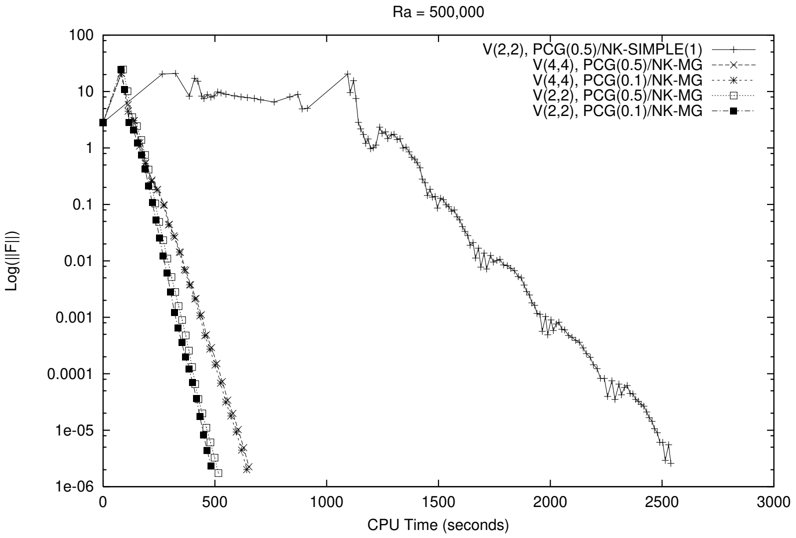

FIG. 3.8. *Convergence histories of FAS for $Ra = 500,000$. Different numbers of pre- and post-smoothing sweeps, different preconditioners in the coarse grid solver, and different tolerances in the smoother are compared. Results were obtained on a 256×256 grid with 2 grid coarsenings. The NK-MG coarse grid solver uses 2 additional grid coarsenings.*

TABLE 3.2
Summary of achieved rates of convergence.

Ra	$20,000$	$50,000$	$100,000$	$500,000$
ρ	0.49	0.52	0.48	0.57

were achieved. In each case, these rates of convergence were achieved with FAS/PCG(δ)/NK-MG.

4. Summary and Conclusions. Classical pressure-correction strategies for solving the steady-state incompressible Navier-Stokes equations can be effectively used as multigrid smoothers. While the smoothing properties of SIMPLE have been previously established, the use of SIMPLER in this context appears to be new. SIMPLER was more effective for the natural convection problem that was studied. A re-examination of conventional strategies for implementing single-grid versions of SIMPLER was found to be necessary to obtain good performance when it is used in as a multigrid smoother. This required use of a highly efficient elliptic solver in order for the SIMPLER smoother to be competetive.

The NK-MG coarse grid solver was found to enhance the robustness of FAS-SIMPLER. This strategy for solving the coarse grid problem allows some flexibility in deciding which grid is the "coarse grid" in the FAS method while still employing the entire grid hierarchy to compute a solution. It was also found that, as the difficulty of the problem increased, it became more important to compute the projection step in SIMPLER to prescribed accuracy in order to obtain acceptable behavior of the NK-MG coarse grid solver. It should be noted that no combination of strategies was found that enabled convergence of FAS-SIMPLER for $Ra = 1,000,000$. Overall, the best performance was achieved through a combination of SIMPLER smoothing in the nonlinear multigrid method with a Newton-Krylov coarse grid solver that used SIMPLE smoothing in its linear multigrid preconditioner.

Acknowledgment. A portion of the computations presented in this paper were per-formed on facilities provided by the University of Utah Center for High Performance Computing.

REFERENCES

[1] J. B. BELL, P. COLELLA, AND H. M. GLAZ, *A second-order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.
[2] M. J. BOOTH, A. G. SCHLIJPER, L. E. SCALES, AND A. D. J. HAYMET, *Efficient solution of liquid state integral equations using the Newton-GMRES algorithm*, Comput. Phys. Comm., (submitted).
[3] A. BRANDT, *Multilevel adaptive solution to boundary value problems*, Math. Comp., 31 (1977), pp. 333–390.
[4] ———, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Rep. GMD-Studien Nr. 85, Gesellschaft für Mathematik und Datenverarbeitung, May 1984.
[5] A. BRANDT AND I. YAVNEH, *Inadequacy of first-order upwind difference schemes for some recirculating flows*, J. Comp. Phys., 93 (1991), pp. 128–143.
[6] P. N. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.
[7] X.-C. CAI, W. D. GROPP, D. E. KEYES, R. G. MELVIN, AND D. P. YOUNG, *Parallel Newton-Krylov-Schwarz algorithms for transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246–265.
[8] X.-C. CAI, W. D. GROPP, D. E. KEYES, AND M. D. TIDRIRI, *Newton-Krylov-Schwarz methods in CFD*, in Proceedings of the International Workshop in the Navier-Stokes Equations, R. Rannacher, ed., Vieweg Verlag, Braunschweig, 1994.
[9] T. F. CHAN, J. XU, AND L. ZIKATANOV, *An agglomeration multigrid method for unstructured grids*, in Domain Decomposition Methods 10, J. Mandel, C. Farhat, and X.-C. Cai, eds., no. 218 in Contemporary Mathematics, American Mathematical Society, 1998, pp. 67–81.
[10] A. J. CHORIN, *Numerical solution of the Navier-Stokes equations*, Math. Comp., 22 (1968), pp. 745–762.
[11] G. DE VAHL DAVIS, *Natural convection of air in a square cavity: a benchmark numerical solution*, Internat. J. Numer. Methods Fluids, 3 (1983), pp. 249–264.
[12] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[13] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.
[14] ———, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.
[15] R. W. FREUND, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., (1993), pp. 470–482.
[16] R. W. FREUND, G. H. GOLUB, AND N. M. NACHTIGAL, *Iterative solution of linear systems*, Acta Numerica, (1992), pp. 87–100.
[17] H. JOHANSEN AND P. COLELLA, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys., 147 (1998), pp. 60–85.
[18] D. A. KNOLL AND P. R. MCHUGH, *An inexact Newton algorithm for solving the tokamak edge plasma fluid equations on multiply connected domains*, J. Comput. Phys., 116 (1995), pp. 281–291.
[19] D. A. KNOLL, P. R. MCHUGH, AND D. E. KEYES, *Newton-Krylov methods for low-mach-number compressible combustion*, AIAA J., 34 (1995), pp. 961–967.
[20] D. A. KNOLL AND W. J. RIDER, *A multigrid preconditioned Newton-Krylov method*, SIAM J. Sci. Comput., (submitted).
[21] B. KOOBUS, M.-H. LALLEMAND, AND A. DERVIEUX, *Unstructured volume-agglomeration MG: Solution of the Poisson equation*, Internat. J. Numer. Methods Fluids, 18 (1994), pp. 27–42.
[22] D. J. MAVRIPILIS AND V. VENKATAKRISHNAN, *A 3D agglomeration multigrid solver for the Reynolds-averaged Navier-Stokes equations on unstructured meshes*, Internat. J. Numer. Methods. Fluids, 23 (1996), pp. 527–544.
[23] P. R. MCHUGH AND D. A. KNOLL, *Fully coupled finite volume solutions of the incompressible Navier–Stokes and energy equations using an inexact Newton method*, Int. J. Numer. Methods. Fluids, 19 (1994), pp. 439–455.
[24] J. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.
[25] S. V. PATANKAR, *A calculation procedure for two-dimensional elliptic situations*, Numer. Heat Transfer, 4 (1981), pp. 409–425.
[26] S. V. PATANKAR AND D. B. SPALDING, *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows*, Internat. J. Heat Mass Transfer, 15 (1972), pp. 1787–1806.
[27] M. PERNICE, *A multigrid preconditioned Newton-Krylov method for the steady-state incompressible Navier-Stokes equations*, (in preparation).

[28] M. PERNICE AND H. F. WALKER, *NITSOL: a Newton iterative solver for nonlinear systems*, SIAM J. Sci Comput., 19 (1998), pp. 302–318.

[29] M. RAW, *A coupled algebraic multigrid method for the 3D Navier-Stokes equations*, in Fast solvers for flow problems, W. Hackbusch and G. Wittum, eds., no. 49 in Notes on Numerical Fluid Mechanics, Vieweg, 1995, pp. 204–215.

[30] Y. SAAD, *ILUT: A dual threshold incomplete LU factorization*, Numer. Linear Algebra. Appl., 1 (1994), pp. 387–402.

[31] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[32] J. N. SHADID, R. S. TUMINARO, AND H. F. WALKER, *An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport*, J. Comp. Phys., 137 (1997), pp. 155–185.

[33] G. J. SHAW AND S. SIVALOGANATHAN, *On the smoothing properties of the SIMPLE pressure-correction algorithm*, Internat. J. Numer. Methods. Fluids, 8 (1988), pp. 441–461.

[34] S. SIVALOGANATHAN AND G. J. SHAW, *A multigrid method for recirculating flows*, Internat. J. Numer. Methods. Fluids, 8 (1988), pp. 417–440.

[35] H. A. VAN DER VORST, *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[36] J. P. VAN DOORMAAL AND G. D. RAITHBY, *Enhancements of the SIMPLE method for predicting incompressible fluid flows*, Numer. Heat Transfer, 7 (1984), pp. 147–163.

[37] S. P. VANKA, *Block implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.

[38] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley and Sons, Inc., 1991.

[39] G. WITTUM, *Multigrid methods for Stokes and Navier-Stokes equations*, Numer. Math., 54 (1989), pp. 543–563.