MULTIGRID SOLUTION OF THE NAVIER-STOKES EQUATIONS ON HIGHLY STRETCHED GRIDS WITH DEFECT CORRECTION

Peter M. Sockol
Internal Fluid Mechanics Division
NASA Lewis Research Center
Cleveland, OH 44135, U.S.A.

SUMMARY

Relaxation-based multigrid solvers for the steady incompressible Navier-Stokes equations are examined to determine their computational speed and robustness. Four relaxation methods with a common discretization have been used as smoothers in a single tailored multigrid procedure. The equations are discretized on a staggered grid with first order upwind used for convection in the relaxation process on all grids and defect correction to second order central on the fine grid introduced once per multigrid cycle. A fixed W(1,1) cycle with full weighting of residuals is used in the FAS multigrid process. The resulting solvers have been applied to three 2D flow problems, over a range of Reynolds numbers, on both uniform and highly stretched grids. In all cases the L_2 norm of the velocity changes is reduced to 10^{-6} in a few 10's of fine grid sweeps. The results from this study are used to draw conclusions on the strengths and weaknesses of the individual relaxation schemes as well as those of the overall multigrid procedure when used as a solver on highly stretched grids.

1. INTRODUCTION

In recent years there has been considerable progress in the development of multigrid solvers for the steady incompressible Navier-Stokes equations. The multigrid process and its application to fluid dynamics has been well described by Brandt[1]. Ghia *et al.*[2] used the streamfunction vorticity formulation with the coupled strongly implicit scheme of Rubin and Khosla[3] as a smoothing operator and an accommodative multigrid cycle. Defect correction was used to increase the accuracy of the convection terms. Vanka[4] employed a locally coupled Gauss-Seidel smoother for the primitive variable formulation together with an accommodative cycle. Demuren[5] extended Vanka's smoother to one in which local corrections were coupled to neighboring pressure corrections and solved the resulting equations by both a strongly implicit technique and an alternating direction line Gauss-Seidel scheme. Thompson and Ferziger[6] used Vanka's smoother as well as a fully coupled alternating direction line Gauss-Seidel extension and an accommodative cycle. This study also introduced defect correction together with local adaptive grid refinement. Sivaloganathan and Shaw[7] used the SIMPLE pressure-correction scheme of Patankar and Spalding[8] as a smoother for the primitive variable formulation. The smoothing analysis given in Shaw and Sivaloganathan[9] indicates that a fixed V-cycle was used in the multigrid process. Dick[10] developed a partially flux-split discretization for the primitive variable formulation and used a coupled red-black smoother and a fixed W-cycle. Finally, a few solvers have used boundary-fitted curvilinear coordinates with primitive variables. Joshi and Vanka[11] extended Vanka's coupled Gauss-Seidel relaxation technique to this system. Rayner[12] and Shyy *et al.*[13] developed variants to the SIMPLE pressure correction method for use as smoothers with the latter applicable to all speeds. The last three references all employed a fixed V-cycle.

In most of the above efforts a single relaxation scheme has been used as a smoothing operator in a chosen multigrid cycle and applied to one or more problems in order to demonstrate the characteristics of the flow solver. This doesn't provide much guidance in the choice of smoother or multigrid cycle for the developer of a solver for a particular application. Furthermore, among the above works only Brandt[1] and Thompson and Ferziger[6] have addressed the need for highly refined grids in local regions, which is present in most flow problems. The adaptive use of several levels of uniform local subgrids[6] is attractive in the multigrid context since it adds extra points only where they are needed. A more conventional approach employs stretched grids which may make it easier to resolve thin regions of steep gradients such as boundary layers adjacent to solid surfaces. This raises the question, however, as to whether fast multigrid performance can be maintained on these grids.

The present work considers the primitive variable formulation of the steady incompressible Navier-Stokes equations in Cartesian coordinates. Four relaxation methods with a common discretization have been used as smoothers and embedded in a single tailored multigrid procedure. The equations are discretized on a staggered grid with first order upwind used for convection in the relaxation process on all grids and defect correction to second order central on the fine grid introduced once per multigrid cycle. The resulting solvers have been applied to three two-dimensional problems over a range of Reynolds numbers on both uniform and highly stretched grids. The results from this study are used to draw conclusions on the strengths and weaknesses of the individual relaxation schemes as well as those of the overall multigrid procedure when used as a solver on highly stretched grids. The results from an earlier study using first order hybrid differencing will be presented elsewhere[14] in somewhat greater detail.

2. DISCRETE FORMULATION

The steady incompressible Navier-Stokes equations in non-dimensional form are written as

$$\frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{1}$$

$$\frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \tag{2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{3}$$

where u and v are the x and y velocity components, p is the pressure, and Re is the Reynolds number.

These equations are discretized on a staggered grid using a finite volume approach:

$$
\begin{aligned}
-R^u_{i,j} &\equiv L^u\, u_{i,j} + dy_j\, \Delta_x p_{i,j} &= 0, \qquad (4)\\
-R^v_{i,j} &\equiv L^v\, v_{i,j} + dx_i\, \Delta_y p_{i,j} &= 0, \qquad (5)\\
-R^c_{i,j} &\equiv dy_j\, \nabla_x u_{i,j} + dx_i\, \nabla_y v_{i,j} &= 0, \qquad (6)
\end{aligned}
$$

where Δ_x, ∇_x, Δ_y, ∇_y, are forward and backward differences in x and y, respectively, $dx_i = x_i - x_{i-1}$, $dy_j = y_j - y_{j-1}$, and

$$
\begin{aligned}
L^u\, u_{i,j} &= a^u_c\, u_{i,j} - a^u_w\, u_{i-1,j} - a^u_e\, u_{i+1,j} - a^u_s\, u_{i,j-1} - a^u_n\, u_{i,j+1}, \qquad (7)\\
L^v\, v_{i,j} &= a^v_c\, v_{i,j} - a^v_w\, v_{i-1,j} - a^v_e\, v_{i+1,j} - a^v_s\, v_{i,j-1} - a^v_n\, v_{i,j+1}, \qquad (8)
\end{aligned}
$$

When these expressions require points outside the domain, such as $L^u\, u_{i,j}$ adjacent to a horizontal boundary, these points are transferred to the boundary by quadratic extrapolation. A linear extrapolation is employed at an outflow boundary where $p_{i,j}$ is specified.

The coefficients in Eqs. (7) and (8) are obtained by utilizing first order upwinding for convection and second order central differencing for diffusion. The difference between first order upwind and second order central convection discretizations on the finest grid is added as a defect correction source term in a manner similar to that of Thompson and Ferziger[6]. Prior to each sweep through the grid a single set of coefficients, a^p, is obtained for equations centered on the $p_{i,j}$ locations and held constant during the sweep. The coefficients a^u and a^v are obtained by averaging. Thus

$$
(a^u_c)_{i,j} = [(a^p_c)_{i,j} + (a^p_c)_{i+1,j}]/2, \qquad (a^v_c)_{i,j} = [(a^p_c)_{i,j} + (a^p_c)_{i,j+1}]/2.
$$

For the convective terms this is equivalent to obtaining the cell face velocities by averaging. For the viscous terms this introduces an error on a stretched grid that is of the same order as the truncation error. In the immediate vicinity of a reentrant corner this practice must be modified to ensure that the convective velocity normal to the wall is set to zero.

3. RELAXATION METHODS

Each of the relaxation methods employed as a multigrid smoother in this work is adapted from or similar to a known technique from the literature, and hence the descriptions of the schemes will be brief. The methods are written in a common block-tridiagonal form for the corrections along a horizontal line:

$$
-\mathbf{A}_i\, \Delta \mathbf{V}_{i-1} + \mathbf{B}_i\, \Delta \mathbf{V}_i - \mathbf{C}_i\, \Delta \mathbf{V}_{i+1} = \mathbf{D}_i, \qquad (9)
$$

where $\Delta \mathbf{V}_i$ is the vector of local corrections, $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ are square matrices, and $\mathbf{D}_i$ is the vector of local residuals. By appropriate choices of the square matrices, Eq.(9) can be used to describe both point or explicit schemes and semi or fully implicit schemes. This equation is now particularized for each of the methods.

The first method, here labeled Block Gauss-Seidel (BGS), is a locally coupled explicit scheme introduced by Vanka[4]. Four discrete momentum equations and one continuity equation are solved for a set of local corrections. In this case

$$\begin{aligned}\Delta\mathbf{V}_{\mathbf{i}} &= (\Delta u_{i-1,j}, \Delta u_{i,j}, \Delta v_{i,j-1}, \Delta v_{i,j}, \Delta p_{i,j})^T,\\ \mathbf{D}_{\mathbf{i}} &= (R^u_{i-1,j}, R^u_{i,j}, R^v_{i,j-1}, R^v_{i,j}, R^c_{i,j})^T,\end{aligned} \tag{10}$$

$\mathbf{B}_{\mathbf{i}}$ is a 5 × 5 matrix,

$$\mathbf{B}_{\mathbf{i}} = \begin{pmatrix} (a^u_c)_{i-1,j} & 0 & 0 & 0 & dy_j \\ 0 & (a^u_c)_{i,j} & 0 & 0 & -dy_j \\ 0 & 0 & (a^v_c)_{i,j-1} & 0 & dx_i \\ 0 & 0 & 0 & (a^v_c)_{i,j} & -dx_i \\ -dy_j & dy_j & -dx_i & dx_i & 0 \end{pmatrix}, \tag{11}$$

and $\mathbf{A}_{\mathbf{i}} = \mathbf{C}_{\mathbf{i}} = \mathbf{0}$. Elimination of the Δu's and Δv's gives a simple expression for $\Delta p_{i,j}$ and back substitution then gives the local Δu's and Δv's. In a single sweep through the grid, each momentum equation is updated twice and each continuity equation once.

The second method, labeled Pressure-linked Line Block Gauss-Seidel (PLBGS), is a locally coupled semi-implicit scheme which is similar to the line relaxation scheme of Demuren[5]. This case is a simple extension of BGS:

$$\begin{aligned}\Delta\mathbf{V}_{\mathbf{i}} &= (\Delta u_{i,j}, \Delta v_{i,j-1}, \Delta v_{i,j}, \Delta p_{i,j})^T,\\ \mathbf{D}_{\mathbf{i}} &= (R^u_{i,j}, R^v_{i,j-1}, R^v_{i,j}, R^c_{i,j})^T,\end{aligned} \tag{12}$$

$\mathbf{B}_{\mathbf{i}}$ is a 4 × 4 matrix obtained by eliminating the top row and left column from Eq.(11), and $\mathbf{A}_{\mathbf{i}} = \mathbf{C}_{\mathbf{i}} = \mathbf{0}$ except for the lower left and upper right corner elements, respectively. Elimination of the Δu's and Δv's gives a scalar tridiagonal equation for the Δp's along the horizontal line and back substitution then gives the Δu's and Δv's along the line. During a single sweep in the $+y$ direction, each u-momentum equation is updated once, each v-momentum equation twice, and each continuity equation once. The fewer momentum updates and the efficiency of the scalar tridiagonal inversion gives a scheme that costs 15% less per sweep than BGS. In general both x and y sweeps are combined in an alternating pattern to form an effective relaxation technique.

The third method, labeled Line Block Gauss-Seidel (LBGS), is a locally coupled, fully implicit scheme, which is apparently very similar to the coupled alternating line approach of Thompson and Ferziger[6]. The vectors $\Delta\mathbf{V}_{\mathbf{i}}$ and $\mathbf{D}_{\mathbf{i}}$ and the matrix $\mathbf{B}_{\mathbf{i}}$ are the same as for PLBGS, while $\mathbf{A}_{\mathbf{i}}$ and $\mathbf{C}_{\mathbf{i}}$ are 4 × 4 matrices having diagonal plus the lower left and upper right corner elements, respectively. The number of equation updates and sweeping patterns are the same as for PLBGS. In this case algebraic elimination in the block-tridiagonal inversion gives a scheme that costs only 15% more per sweep than BGS.

The final method is the Semi-Implicit Pressure-Correction scheme (SIMPLE) introduced by Patankar and Spalding[8]. In this case

$$\begin{aligned}\Delta\mathbf{V}_{\mathbf{i}} &= (\Delta u_{i,j}, \Delta v_{i,j})^T,\\ \mathbf{D}_{\mathbf{i}} &= (R^u_{i,j}, R^v_{i,j})^T,\end{aligned} \tag{13}$$

where $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$ are diagonal 2×2 matrices. The pressure is obtained from an elliptic equation derived by substituting reduced forms of the discrete momentum equations for coupled velocity and pressure corrections into continuity. For this work one SIMPLE iteration consists of a single scalar line Gauss-Seidel sweep for each momentum equation with the pressure fixed. This is followed by four alternating direction line Gauss-Seidel sweeps of the elliptic pressure-correction equation. Taking more than one sweep through the momentum equations before correcting the pressure *invariably* resulted in partial decoupling of the velocity components and slower convergence. Each of these combined SIMPLE iterations costs about 30% more than one sweep of BGS.

For each of these relaxation techniques some degree of underrelaxation is required to obtain convergence. In the present work this is implemented through direct modification of the momentum equations. For BGS, LBGS, and SIMPLE, the diagonal velocity coefficients, a^u_c and a^v_c, in the matrix $\mathbf{B}_i$ are divided by a factor r_{mom} where $0 < r_{mom} < 1$. For PLBGS the residuals, R^u and R^v, are multiplied by r_{mom}. In addition for SIMPLE the pressure corrections and the corresponding velocity corrections required to satisfy continuity are unrelaxed.

Finally, we note that considerable improvement can be obtained with each of the above methods by employing a symmetric sweeping pattern. Thus for BGS each lexicographic sweep is followed by one in the reverse direction. For PLBGS, LBGS, and SIMPLE a four sweep symmetric alternating line pattern is used, i.e. relaxation is performed sequentially in the $+x$, $+y$, $-y$, $-x$ directions. These techniques result in an approximately 25% improvement in convergence rates.

4. MULTIGRID IMPLEMENTATION

Local relaxation methods, such as those of the previous section, are in general much more efficient at reducing short wavelength error components on a given grid than those of longer wavelength. Multigrid seeks to overcome this problem by transferring the longwave components of the solution to a sequence of coarser grids where relaxation is more effective and much cheaper. Since the FAS-FMG technique used in this work has been well documented in the literature[1,2,4–7], it will not be described here. The focus will instead be on the current implementation and in particular on those aspects which are important for achieving a fast, robust Navier-Stokes solver.

In the present work the coarse grids are created by "standard coarsening," i.e., every second grid point in both x and y is deleted from one grid to the next coarser grid. The fine-to-coarse restriction operator $\hat{I}^c_f$ for unknowns employs cell-face averaging for the velocities,

$$u^c_{i,j} = (u_{i,j-1}dy_{j-1} + u_{i,j}dy_j)/dy^c_j, \qquad v^c_{i,j} = (v_{i-1,j}dx_{i-1} + v_{i,j}dx_i)/dx^c_i, \tag{14}$$

and full-weighting for the pressures,

$$\begin{aligned} p^c_{i,j} = (p_{i-1,j-1}dx_{i-1}dy_{j-1} + p_{i-1,j}dx_{i-1}dy_j \\ + p_{i,j-1}dx_i dy_{j-1} + p_{i,j}dx_i dy_j)/(dx^c_i dy^c_j), \end{aligned} \tag{15}$$

where ( )^c represents a coarse-grid value. The restriction operator I_f^c for residuals uses full-weighting, in which all the fine-grid contributions to a coarse-grid cell are accounted for:

$$\begin{aligned}(I_f^c R^u)_{i,j} &= R^u_{i,j-1} + R^u_{i,j} + \frac{1}{2}(R^u_{i-1,j-1} + R^u_{i-1,j} + R^u_{i+1,j-1} + R^u_{i+1,j}),\\ (I_f^c R^v)_{i,j} &= R^v_{i-1,j} + R^v_{i,j} + \frac{1}{2}(R^v_{i-1,j-1} + R^v_{i,j-1} + R^v_{i-1,j+1} + R^u_{i,j+1}),\end{aligned} \tag{16}$$

where R^u and R^v, given by Eqs.(4) and (5), are already area-weighted. With the cell-face averaging given by Eqs.(14) and full-weighting similar to Eq.(15) for R^c, as defined by Eq.(6), the coarse-grid source term vanishes for the continuity equation.

In many of the previous works[4,5,7,13] cell-face averaging was also used in the restriction of R^u and R^v. For uniform grids this has little effect on the multigrid convergence rate. For the highly stretched grids employed in this work this proved to be ineffective. In some cases convergence slowed by a factor of three or four. In others little or no benefit was gained from the multigrid process.

The coarse-to-fine prolongation operator I_c^f for corrections employs bilinear interpolation in computational space where the grid spacing is taken to be uniform. For fine grid points adjacent to boundaries, a zero normal gradient is assumed for pressures. The overall convergence has proven to be insensitive to the details of this approximation. The same operator with one modification is also used to interpolate "converged" results to obtain initial values on a fine grid in the FMG process. The velocity component parallel to an adjacent wall is obtained by bilinear extrapolation from the interior since the boundary layer is poorly resolved on the coarse grid.

The multigrid solvers in this work have been coded to permit fixed V-cycles and W-cycles. During the course of this effort it was found that for the difficult cases with high Reynolds numbers or highly stretched grids a W(1,1) cycle was the most effective strategy in terms of robustness and computational cost. Hence, all results presented in this paper were performed using this cycle. The defect correction source term, discussed earlier, is updated once per cycle on the finest grid. Accommodative cycles[1,2,4–6], which decide on whether or not to restrict to a coarser grid based on the ratio of errors from two successive sweeps, proved to be too costly since the second sweep on each visit to a grid contributed little to the overall convergence of the method.

The symmetric sweeping pattern described in the previous section has been interleaved with the multigrid process. A sweep counter is established for every grid level, and on each visit to that level the next direction in the sweep pattern for that grid is performed. This proved to be sufficient to give all the convergence benefits of the sweeping symmetry. Finally, it should be noted that varying the momentum relaxation factor r_{mom} from grid to grid during the cycle provided considerable performance enhancement for the BGS, PLBGS, and LBGS solvers. No benefit, however, was observed when this was tried with the SIMPLE-based solver.

5. CONVERGENCE CRITERIA

The various convergence criteria used in this work are all based on an L_2 norm of the dynamic velocity changes ocurring during a sweep through the grid. This would seem to be a more appropriate form for a system of coupled equations than one based on a combination of the residuals of the different equations. The pressures have been excluded since they are only determined to within an arbitrary constant. Introduce the definition

$$\varepsilon^k = \sqrt{\sum_{i,j=1}^{n_x^k, n_y^k} [(\Delta u_{i,j}^k)^2 + (\Delta v_{i,j}^k)^2]/(2n_x^k n_y^k)}, \tag{17}$$

where n_x^k and n_y^k are the number of cells on grid k in x and y respectively, and $\Delta u_{i,j}^k$, $\Delta v_{i,j}^k$ are the dynamic velocity changes obtained during a sweep on grid k. Then for a sequence of coarse-to-fine grids, k = 1 to m, the overall convergence criterion on grid m is taken as

$$\varepsilon^m < 10^{-6}. \tag{18}$$

In most cases at convergence given by Eq.(18) the value of $\max(\Delta u, \Delta v)$ is approximately 10^{-5}. For intermediate grids in the FAS-FMG process, convergence before interpolating to the next finer grid is taken as

$$\varepsilon^k < 10^{-3}, \tag{19}$$

and for the coarsest grid, k = 1, "solution" is given by

$$\varepsilon^1 < \varepsilon^k/10, \tag{20}$$

where now ε^k is the most recent error on the current finest grid.

Finally, it is noted that all computations in this work were performed on an Amdahl 5980 in scalar mode. All CPU times reported in the next sections are for this machine.

6. COMPUTATIONAL RESULTS

Three problems have been chosen to test the performance of the multigrid solvers under different conditions: flow in a driven cavity, developing flow in a straight channel, and flow over an open cavity.

Driven Cavity Flow

The driven cavity is the prototypical recirculating flow and has long been used as a standard test problem for Navier-Stokes solvers. The second-order streamfunction-vorticity results of Ghia *et al.*[2] are generally accepted as the standard. Flow is set up in a square cavity with three stationary walls and a top lid that moves to the right with constant speed ($u = 1$). Profiles of u on the vertical centerline computed on a uniform 256×256 grid for $Re = 1000$ and 5000 are compared with the standard results[2] in Figure 1. The present defect correction results agree with the standard to within plotting accuracy.

The first set of results for this flow is for a uniform grid with Re varying from 100 to 5000. Table I compares the uniform grid results for each solver on a 256×256 grid in terms of cpu times, number of fine grid sweeps, and total work units for each case where a work unit is the cpu time required for one fine grid sweep of the particular smoother. Here

r^{fg}_{mom} is the fine grid relaxation factor for the solver. As the Reynolds number increases the table indicates a significant advantage for BGS and PLBGS over LBGS and SIMPLE due to faster convergence and less cost per sweep.

The second set of results is obtained for $Re = 1000$ on a grid with hyperbolic tangent stretching in x and y and the maximum mesh aspect ratio (AR) varying from 1 to 40. Table II compares the stretched grid results for each solver on a 256×256 grid. As AR is increased to large values BGS is seen to have a significant advantage over the other three methods in both number of sweeps and cpu time. The use of highly stretched grids produces a strong asymmetry in the momentum equation coupling coefficients in regions of high mesh aspect ratio and this was expected to adversely affect the smoothing properties of an explicit scheme[1] such as BGS. The alternating direction semi-implicit and fully implicit schemes were introduced to see if they would give more robust performance for these cases. This proved not to be true for the Navier-Stokes solvers used in this study.

Developing Channel Flow

The second test problem is the deceptively simple one of developing flow in a straight channel one unit high by four units long. Uniform velocities ($u = 1$, $v = 0$) are specified at the entrance and a constant pressure ($p = 0$) is set at the exit. Note, for incompressible flow, the common exit condition, $\partial u/\partial x = 0$, implies $\partial v/\partial y = \partial p/\partial y = 0$. Profiles of u vs y along the channel for $Re = 1000$ are shown in Figure 2. For this and higher Reynolds numbers the flow is far from fully developed at the exit. This flow has velocities strongly aligned with the x direction over much of the domain and the u momentum equation becomes increasingly decoupled in y away from the walls as Re is increased. This situation is known to cause problems for multigrid solvers (see e.g. Brandt[1] and Mulder[15]) and thus was chosen as a fitting test case for this study.

The first set of results is for a uniform grid with Re again varying from 100 to 5000. The uniform grid results for each solver on a 256×64 grid are compared in Table III. It is evident that the multigrid performance of all solvers degrades more rapidly with increasing Re than was the case for the driven cavity. The relatively poor performance of SIMPLE is probably due to the partial decoupling between u and v at high Re which was observed during the iterative process. Note, however, that all methods still converged in under 100 fine grid sweeps even at the highest Reynolds numbers.

The second set of results for this flow is for hyperbolic tangent stretching in y only, again with AR varying from 1 to 40 and $Re = 1000$. Stretched grid results for each solver on a 256×64 grid are compared in Table IV. As AR is increased to large values, it is evident that LBGS has a major advantage over the other smoothers in both fine grid sweeps and cpu time. This case of strong alignment on a stretched grid is the only one in which an implicit scheme (LBGS) has a substantial advantage over the explicit BGS.

Open Cavity Flow

The final test problem combines the driven cavity and developing channel flows and adds the complication of a strong corner singularity. The domain consists of a channel one unit high by two units long on top of an open square cavity one unit on a side located at the left boundary of the channel. Uniform flow (u = 1, v = 0) enters the channel at the left and the flow exits at the right (p = 0). Streamfunction and vorticity contours for Re = 1000 are shown in Figure 3. Note the lack of separation and the strong concentration of vorticity contours at the downstream corner.

As before the first set of results is for a uniform grid with Re varying from 100 to 5000. The uniform grid results for each solver on a 128×128 + 256×128 grid are compared in Table V. The results for both fine grid sweeps and cpu time show that BGS, PLBGS, and LBGS remain competitive as Reynolds number is increased but SIMPLE suffers a substantial penalty.

The second set of results for this flow is for hyperbolic tangent stretching in both x and y, in each of three square regions, with AR varying from 1 to 40 and Re = 1000. The stretched grid results for each solver on a 128×128 + 256×128 grid are compared in Table VI. Here it is evident that BGS has a significant advantage in fine grid sweeps and cpu time as AR increases. It should also be noted that PLBGS and LBGS appeared to be more sensitive to the presence of the corner singularity and to the choice of r_{mom} for the set of grids used in the multigrid process. However no detailed study of this effect was performed.

7. CONCLUSIONS

From the above results, it is evident that a proper combination of tailored multigrid elements can yield a fast robust solver for the steady incompressible Navier-Stokes equations even on highly stretched grids. In particular, for fine-to-coarse restriction of residuals, the use of full weighting is important on stretched grids. For coarse-to-fine prolongation of corrections, on the other hand, bilinear interpolation works well and is insensitive to the details of the boundary treatment. And finally a fixed W(1,1) multigrid cycle appears to offer a good mix of robustness and computational efficiency.

For recirculating flows such as the driven cavity, all four smoothers are effective and competitive. On uniform grids BGS and PLBGS offer a significant advantage over LBGS and SIMPLE, primarily due to less cost per sweep. On stretched grids BGS and SIMPLE show superior multigrid performance, but BGS is substantially cheaper per sweep.

For strongly aligned flows such as that in a developing channel, all four solvers degrade more rapidly with increasing Reynolds number than for recirculating flows with SIMPLE falling off much more rapidly than the others, but they all still converge in under 100 fine grid sweeps. On highly stretched grids, however, LBGS offers a major advantage in both multigrid performance and net cpu time over the other three smoothers. This is the only case in which an implicit scheme is distinctly superior to the explicit BGS.

For mixed recirculating/aligned flows such as the open cavity, all four smoothers are effective. On uniform grids, SIMPLE again degrades much more rapidly than the others with increasing Reynolds number. On stretched grids BGS offers a small advantage in multigrid performance, but this becomes significant when net cpu time is considered. It is also notable that BGS is less sensitive than the other smoothers to the corner singularity in this flow.

On balance BGS offers the best mix of robustness and computational speed for all three classes of flows. The semi-implicit schemes PLBGS and SIMPLE offer little or no advantage and in general are less robust. The fully implicit LBGS is superior only for the case of highly aligned flows on stretched grids. The pressure correction scheme SIMPLE is in general more costly than the other three and degrades much more rapidly than the others with increasing Reynolds number. Finally, we note that for a general multigrid solver set up using domain decomposition, it might be highly effective to use BGS over most domains but retain the option to use LBGS in strongly aligned domains.

REFERENCES

1. A. Brandt, 'Multigrid Techniques: 1984 Guide With Applications to Fluid Dynamics', von Karman Institute, *Lecture Series 1984-04*, 1984.
2. U. Ghia, K. N. Ghia and C. T. Shin, 'High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method', *J. Comput. Phys.,* **48**, 387-411 (1982).
3. S. G. Rubin and P. K. Khosla, 'Navier-Stokes Calculations with a Coupled Strongly Implicit Method—I', *Comput. Fluids,* **9**, 163-180 (1981).
4. S. P. Vanka 'Block-Implicit Multigrid Solution of the Navier-Stokes Equations in Primitive Variables', *J. Comput. Phys.,* **65**, 138-158 (1986).
5. A. O. Demuren 'Application of Multi-Grid Methods for Solving the Navier-Stokes Equations', *Proc. Inst. Mech. Engrs., Part C.,* **203**, 255-265 (1989).
6. M. C. Thompson and J. H. Ferziger 'An Adaptive Multigrid Technique for the Incompressible Navier-Stokes Equations', *J. Comput. Phys.,* **82**, 94-121 (1989).
7. S. Sivaloganathan and G. J. Shaw 'A Multigrid Method for Recirculating Flows', *Int. J. Numerical Methods Fluids,* **8**, 417-440 (1988).
8. S. V. Patankar and D. B.Spalding 'A Calculation Procedure for Heat and Mass Transfer in 3-D Parabolic Flows', *Int. J. Heat Mass Transfer,* **15**, 1787-1806 (1972).
9. G. J. Shaw and S. Sivaloganathan 'On the Smoothing Properties of the SIMPLE Pressure-Correction Algorithm', *Int. J. Numerical Methods Fluids,* **8**,441-461 (1988).
10. E. Dick 'A Multigrid Method for Steady Incompressible Navier-Stokes Equations Based on Partial Flux Splitting', *Int. J. Numerical Methods Fluids,* **9**, 113-120 (1989).
11. D. S. Joshi and S. P. Vanka 'Multigrid Calculation Procedure for Internal Flows in Complex Geometries', *Numer. Heat Transfer, Part B.,* **20**, 61-80 (1991).
12. D. Rayner 'Multigrid Flow Solutions in Complex Two-Dimensional Geometries', *Int. J. Numerical Methods Fluids,* **13**, 507-518 (1991).
13. W. Shyy, M-H. Chen and C-S Sun 'A Pressure-Based FMG/FAS Algorithm for Flow at All Speeds', AIAA 92-0548, 30th Aerospace Sciences Meeting & Exhibit (1992).
14. P. M. Sockol 'Multigrid Solution of the Navier-Stokes Equations on Highly Stretched Grids', to be published in *Int. J. Numerical Methods Fluids.*
15. W. A. Mulder 'A New Multigrid Approach to Convection Problems', *J. Comput. Phys.,* **83**, 303-323 (1989).

Table I. Driven Cavity Convergence
Uniform 256×256 grid, AR = 1

			Re		
Scheme (r^{fg}_{mom})	100	400	1000	3200	5000
	(cpu seconds/fg. sweeps/work units)				
BGS (0.6)	140.2	165.4	197.6	350.7	483.4
	9	10	12	22	30
	22.1	26.0	31.0	55.6	76.0
PLBGS (0.7)	147.3	154.6	207.1	348.1	517.4
	10	11	14	24	36
	26.6	27.6	37.2	62.6	93.3
LBGS (0.7)	180.6	183.4	219.0	435.4	642.8
	10	10	12	24	36
	25.1	25.4	30.5	60.5	89.1
SIMPLE (0.7)	257.3	256.7	307.2	554.4	806.5
	12	12	14	26	38
	27.2	27.4	32.8	59.0	85.9

Table II. Driven Cavity Convergence
Stretched 256×256 grid, Re = 1000

			AR		
Scheme (r^{fg}_{mom})	1	5	10	20	40
	(cpu seconds/fg. sweeps/work units)				
BGS (0.6)	197.8	168.6	168.6	199.9	231.3
	12	10	10	12	14
	31.0	26.4	26.4	31.2	36.1
PLBGS (0.5)	260.5	205.7	211.0	268.0	324.6
	18	14	15	19	23
	47.1	36.9	37.9	47.9	57.9
LBGS (0.9)	220.3	220.5	290.2	395.0	604.2
	12	12	16	22	34
	30.5	30.1	39.5	53.7	81.9
SIMPLE (0.7)	311.8	276.2	304.9	305.4	344.8
	14	13	14	14	16
	32.7	28.4	31.9	31.8	36.0

Table III. Developing Channel Convergence
Uniform 256×64 grid, AR = 1

			Re		
Scheme (r^{fg}_{mom})	100	400	1000	3200	5000
	(cpu seconds/fg. sweeps/work units)				
BGS (0.7)	53.0	64.5	82.4	152.3	203.4
	12	14	18	34	46
	30.1	36.4	46.9	87.0	116.2
PLBGS (0.8)	49.6	80.0	95.2	159.9	218.0
	12	20	24	40	56
	32.5	52.2	62.3	105.2	142.5
LBGS (0.8)	59.0	78.5	78.3	136.1	173.5
	12	16	16	28	36
	30.7	40.5	40.7	70.7	89.4
SIMPLE (0.7)	83.6	124.0	168.9	324.4	418.7
	16	24	32	64	84
	39.9	58.8	79.9	154.3	199.6

Table IV. Developing Channel Convergence
Stretched 256×64 grid, Re = 1000

			AR		
Scheme (r^{fg}_{mom})	1	5	10	20	40
	(cpu seconds/fg. sweeps/work units)				
BGS (0.7)	81.9	139.9	202.1	251.7	268.2
	18	32	46	58	62
	46.9	79.4	113.8	141.1	150.8
PLBGS (0.7)	110.2	117.3	193.8	196.3	230.4
	28	30	50	50	59
	72.4	75.5	124.5	127.2	147.5
LBGS (0.85)	78.4	87.8	105.4	123.9	142.1
	16	18	22	26	30
	40.6	44.7	54.0	63.4	73.3
SIMPLE (0.7)	168.2	142.0	162.0	201.1	261.5
	32	28	32	40	52
	79.9	67.6	76.9	95.1	123.6

Table V. Open Cavity Convergence
Uniform 128×128 + 256×128 grid, AR = 1

			Re		
Scheme (r^{fg}_{mom})	100	400	1000	3200	5000
	(cpu seconds/fg. sweeps/work units)				
BGS (0.7)	173.5 12 29.7	203.1 14 34.6	230.7 16 39.7	381.5 26 65.2	573.3 40 98.7
PLBGS (0.7)	186.2 15 36.4	237.7 19 46.4	262,2 20 51.7	465.6 36 91.2	616.6 48 120.9
LBGS (0.8)	166.5 11 25.7	221.1 14 34.4	256.9 16 40.2	439.4 28 68.7	619.5 40 97.5
SIMPLE (0.7)	243.9 14 32.5	319.5 18 42.6	419.6 24 56.1	705.6 40 93.5	1004.2 58 133.6

Table VI. Open Cavity Convergence
Stretched 128×128 + 256×128 grid, Re = 1000

			AR		
Scheme (r^{fg}_{mom})	1	5	10	20	40
	(cpu seconds/fg. sweeps/work units)				
BGS (0.7)	230.0 16 39.6	175.8 12 30.1	176.7 12 30.2	206.0 14 35.2	233.6 16 40.0
PLBGS (0.5)	367.1 28 70.6	263.4 20 50.8	238.3 18 46.1	245.2 19 47.2	298.3 23 57.1
LBGS (0.95)	224.4 14 35.6	219.8 14 34.5	222.2 14 34.6	283.0 18 44.1	343.0 22 53.5
SIMPLE (0.7)	423.0 24 55.9	289.3 16 38.0	280.8 16 37.2	282.1 16 37.2	379.6 22 50.5

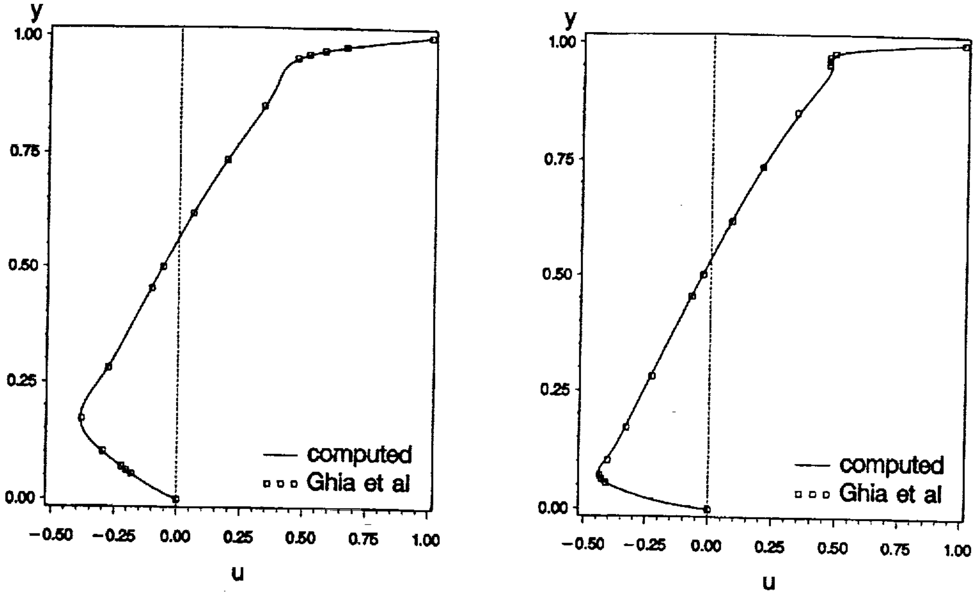

Figure 1. Driven cavity u-velocities on vertical centerline computed on a uniform 256×256 grid for $Re = 1000$ (left) and 5000 (right)

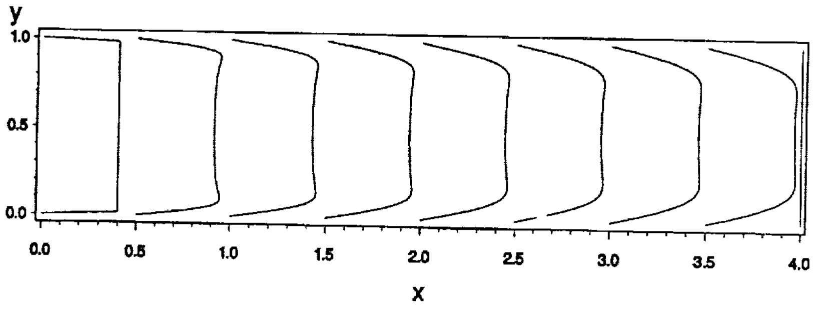

Figure 2. Developing channel u-velocity profiles for $Re = 1000$

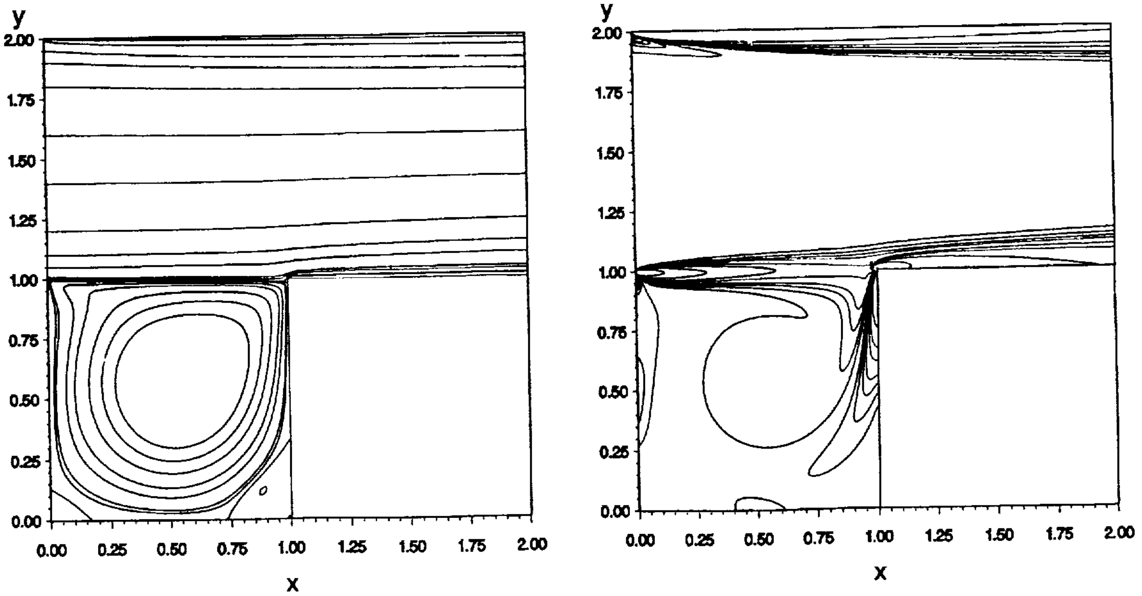

Figure 3. Open cavity stream function (left) and vorticity (right) contours for $Re = 1000$