JOURNAL OF COMPUTATIONAL PHYSICS 65, 138–158 (1986)

Block-Implicit Multigrid Solution of Navier–Stokes Equations in Primitive Variables

S. P. VANKA

Argonne National Laboratory, Components Technology Division, Argonne, Illinois 60439

Received February 20, 1985; revised July 17, 1985

A calculation procedure for rapid computation of steady multidimensional viscous flows is presented. The method solves the Navier–Stokes equations in primitive variables using a coupled block-implicit multigrid procedure. The procedure is applicable to finite-difference formulations using staggered locations of the flow variables. A smoothing technique called symmetrical coupled Gauss–Seidel (SCGS) is proposed and is empirically observed to provide good smoothing rates. The viscous flow in a square cavity with a moving top wall is calculated for a range of Reynolds numbers. Calculations with finite difference grids as large as 321 × 321 nodes have been made to test the accuracy and efficiency of the calculation scheme. The CPU times for these calculations are observed to be significantly smaller than other solution algorithms with primitive variable formulation. The calculated flow fields in the cavity are in good agreement with earlier studies of the same flow situation. © 1986 Academic Press, Inc.

1. INTRODUCTION

In contrast with the vorticity-streamfunction and vorticity-vector potential methods of calculating fluid flows, the primitive variable formulation is the preferred route to the solution of complex fluid flows in engineering geometries. However, the evaluation of the pressure field has always been the difficult issue in the primitive variable approach. In the works of the Los Alamos group, (e.g., Harlow and Welch [1], Harlow and Amsden [2]) time-dependent Navier–Stokes equations are solved in (u, v, p) framework with both explicit and semi-implicit differencing. The explicit method is limited in time-step size by the acoustic Courant number, whereas the semi-implicit method is limited by the velocity Courant number. A staggered mesh system is used in which pressures are stored at cell centers and velocities at cell interfaces.

Another popular series of methods for solving the primitive variable equations was originated by the SIMPLE [4] algorithm of Patankar and Spalding. In SIMPLE, the emphasis has been on solving the steady state equations. Like the Los Alamos methods, the variables are located on a staggered grid and a decoupled solution strategy is used. First, an approximate pressure field is employed to solve the momentum equations. A pressure-correction (p') equation is then derived by

0021-9991/86 $3.00

combining the finite-differenced continuity equation and approximate forms of the momentum equations. The p' equation is solved by alternate line sweeps, and both velocities and pressures are updated to reflect the pressure corrections. Typically, all changes are underrelaxed to obtain numerical stability. SIMPLE has been used in a number of diverse engineering situations (e.g., see [5, 6]). Modifications to SIMPLE to improve the pressure-velocity coupling have been suggested recently by Raithby and Schneider [7], Patankar [8], and Issa [9].

In contrast with the decoupled methodology (also called segregated solution, Raithby and Schneider [7]), a coupled solution of the momentum and continuity equations implicitly retains the pressure-velocity coupling and therefore eliminates the need for the (approximate) pressure or pressure-correction equation. The velocities and pressures are simultaneously updated in a linear sense and iterations are made to remove the nonlinearities. Such techniques are common in the finite-element solution of the fluid flow equations. In Zedan and Schneider [10] and in our own works (Vanka and Leaf [11], Vanka [12, 13]), the equations are solved in the form of blocks consisting of the momentum equations and the continuity equation. Zedan and Schneider [10], however, first derive an approximate pressure equation and employ a coupled SIP algorithm extending the work of Stone [14]. Vanka [12, 13] uses the primitive continuity equation without converting it to a pressure equation. A sparse matrix inversion routine, combined with preordering and domain splitting techniques, was employed to solve the linearized equations. With such a procedure calculations of complex turbulent recirculating and reacting flows have been made in computational times a factor of ten smaller than those required by SIMPLE.

The present study has been motivated by the recent demonstrations of the multigrid technique [15, 16, 17] as an efficient iterative solver for linear and nonlinear elliptic equations. The multigrid technique has several attractive attributes including $O(n)$ operation count in many problems and practically no additional storage arrays over those for the variables and their coefficients. In this study, we investigate the applicability of the multigrid technique for the coupled iterative solution of the momentum and continuity equations. A coupled symmetrical Gauss–Seidel technique is proposed for the smoothing process. The algorithm is applied to the highly nonlinear flow problem in a square cavity with a moving top wall. Calculations are made with finite-difference grids consisting of up to 321 × 321 grid nodes and the results are compared with earlier numerical studies.

2. The Equation Set and Finite Differencing

Planar 2-dimensional flows are governed by the following set of nonlinear partial differential equations:

$$(uu)_x + (vu)_y = -p_x/\rho + \nu u_{xx} + \nu u_{yy}, \tag{1}$$

$$(vu)_x + (vv)_y = -p_y/\rho + \nu v_{xx} + \nu v_{yy}, \tag{2}$$

and

$$u_x + v_y = 0. \tag{3}$$

For simplicity, constant density and constant laminar viscosity are assumed. The above equations are discretized by a hybrid finite-differencing scheme [18], which employs second-order central differencing on both convection and diffusion terms but automatically modifies the convective differencing procedure to an upwind formulation when the local cell Reynolds number exceeds two. The hybrid scheme has the merits of stability at high ($\geqslant 2$) cell Reynolds numbers (albeit of first-order accuracy) and second-order accuracy at low cell Reynolds numbers.

A staggered mesh system has been used, consistent with other works of the primitive variable formulation. Thus velocities are stored on the cell faces and pressures are stored at the cell centers. The resulting finite-difference equations (for uniform grid dimensions of δx and δy) are written as follows;

$$\begin{aligned} A_c^u u_{i+1/2,j} = {} & A_n^u u_{i+1/2,j+1} + A_s^u u_{i+1/2,j-1} + A_e^u u_{i+3/2,j} \\ & + A_w^u u_{i-1/2,j} + (p_{i,j} - p_{i+1,j})/\rho\, \delta x, \end{aligned} \tag{4}$$

$$\begin{aligned} A_c^v v_{i,j+1/2} = {} & A_n^v v_{i,j+3/2} + A_s^v v_{i,j-1/2} + A_e^v v_{i+1,j+1/2} \\ & + A_w^v v_{i-1,j+1/2} + (p_{i,j} - p_{i,j+1})/\rho\, \delta y, \end{aligned} \tag{5}$$

and

$$(u_{i+1/2,j} - u_{i-1/2,j})/\delta x + (v_{i,j+1/2} - v_{i,j-1/2})/\delta y = 0. \tag{6}$$

The superscripts relate to the variables and the subscript indices denote the location of the variables on the finite-difference grid, i being the index in the x direction. The fractional indices refer to the staggered locations of the velocities. The coefficients A_n, A_s, etc. contain averaged velocities and the diffusion coefficient through the following expressions:

$$A_c^u = A_n^u + A_s^u + A_e^u + A_w^u, \tag{7}$$

$$A_c^v = A_n^v + A_s^v + A_e^v + A_w^v, \tag{8}$$

$$A_w^u = \mathrm{AMAX}(|C_{x-}^u|, D_{x-}^u) + C_{x-}^u, \tag{9}$$

$$A_e^u = \mathrm{AMAX}(|C_{x+}^u|, D_{x+}^u) - C_{x+}^u, \tag{10}$$

$$A_s^u = \mathrm{AMAX}(|C_{y-}^u|, D_{y-}^u) + C_{y-}^u, \tag{11}$$

$$A_n^u = \mathrm{AMAX}(|C_{y+}^u|, D_{y+}^u) - C_{y+}^u, \tag{12}$$

$$A_w^v = \mathrm{AMAX}(|C_{x-}^v|, D_{x-}^v) + C_{x-}^v, \tag{13}$$

$$A_e^v = \mathrm{AMAX}(|C_{x+}^v|, D_{x+}^v) - C_{x+}^v, \tag{14}$$

$$A_s^v = \mathrm{AMAX}(|C_{y-}^v|, D_{y-}^v) + C_{y-}^v, \tag{15}$$

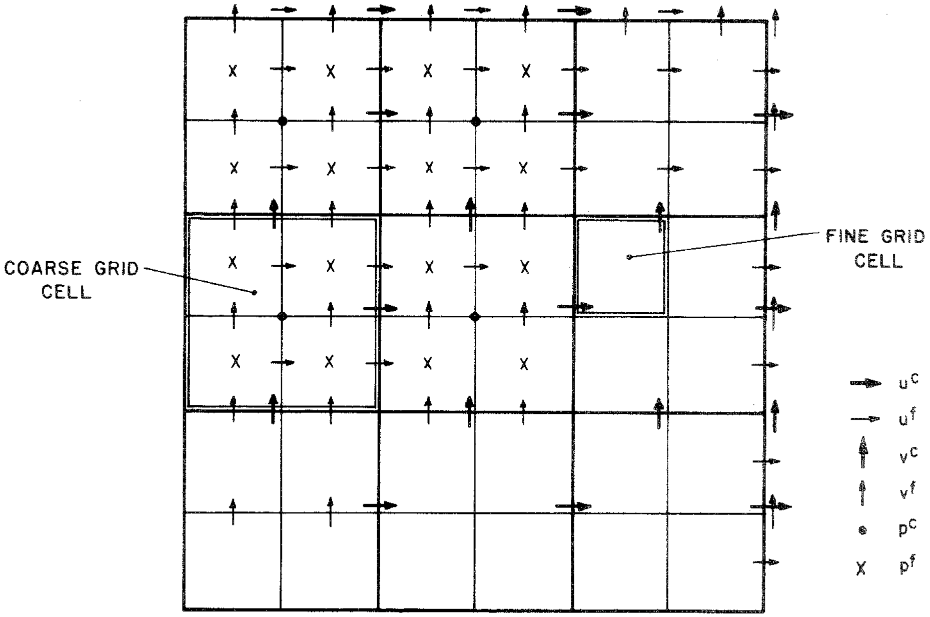


FIG. 1. Staggered mesh arrangement.

and

$$A_n^v = \text{AMAX}(|C_{y+}^v|, D_{y+}^v) - C_{y+}^v, \tag{16}$$

where

$$C_{x-} = 0.5u_{x-}/\delta x; \qquad D_{x-} = \nu/\delta x^2, \tag{17}$$

$$C_{x+} = 0.5u_{x+}/\delta x; \qquad D_{x+} = \nu/\delta x^2, \tag{18}$$

$$C_{y-} = 0.5v_{y-}/\delta y; \qquad D_{y-} = \nu/\delta y^2, \tag{19}$$

and

$$C_{y+} = 0.5v_{y+}/\delta y; \qquad D_{y+} = \nu/\delta y^2. \tag{20}$$

u_{x-}, u_{x+}, v_{y-}, and v_{y+} denote velocities at the $x-$, $x+$, $y-$, $y+$ sides of the cells surrounding the variable in question. δx, δy are dimensions of the finite difference cells (Fig. 1).[1] The AMAX function ensures the central difference operator at cell Re $\leqslant 2$ and changes the differencing to upwinding at cell Re $\geqslant 2$. Also, the diffusion fluxes at the interfaces are made zero on the presumption of a zero derivative at the interface. Equations (7) and (8) are obtained by invoking the mass continuity for

[1] Some minor modifications are necessary for the diffusion terms at the near boundary nodes.

the velocity cells. The above finite difference equations have been obtained by integrating over cells surrounding a local variable, but they are written here as difference operators for convenience in using the multigrid technique.

3. The Multigrid Technique

Given a set of linear finite-difference equations

$$L^k W^k = F^k \tag{21}$$

for a general elliptic equation, any iterative procedure such as Gauss–Seidel, Jacobi, incomplete LU factorization, etc. is known to converge rapidly for the first few iterations and very slowly thereafter. A Fourier analysis of the error reduction process shows that these conventional iterative procedures are most efficient in smoothing out the errors of wavelengths comparable to the mesh size, but are inefficient in annihilating low frequency components. However, the low frequencies on any grid are relatively larger on grids that are coarser than the grid in question. The multigrid technique is based on the premise that each frequency range of error must be smoothed on the grid where it is most suitable to do so. Consequently, the multigrid technique cycles between coarser and finer grids until all the frequency components are appropriately smoothed. The multigrid concept is distinct from the philosophy of starting a fine grid solution from an interpolated coarse grid converged solution. In the latter concept, only a better starting guess is provided. Therefore the starting residual is smaller than a "raw" guess, but the asymptotic rate of convergence is not improved. The multigrid method, on the other hand, cycles between a hierarchy of computational grids D^k with corresponding grid functions W^k, $k = 1, 2,..., M$. The step size on D^k is h_k, and $h_{k+1} = \frac{1}{2} h_k$, so that as k decreases, D^k becomes coarser.

In the simplest version of the multigrid technique [15], the solution is initiated on the finest grid M. A few iterations are performed on grid M until the relaxation procedure (iterations) fails to smooth the residuals at the desired theoretical rate. The iterations are stopped on grid M and the residuals ($L^k w^k - F^k = R^k$; w^k is an approximation to W^k) are transferred to the next coarser grid, obtained typically by doubling the mesh size. The grid functions (variables) are similarly restricted to the coarse grid. In this simplest form of the multigrid (MG) technique, called the correction scheme (CS), a correction function w^{k-1} is calculated by solving the system

$$L^{k-1} w^{k-1} = I_k^{k-1} R^k, \tag{22}$$

where L^{k-1} is the operator on grid $(k-1)$ and I_k^{k-1} is the operator to restrict (interpolate) the residual on grid k. The solution to Eq. (22) can be obtained on grid $(k-1)$ itself if $(k-1)$ is not a large grid. Otherwise a few iterations on $(k-1)$

are performed, after which R^{k-1} is restricted to grid $(k-2)$. When an accurate solution to Eq. (22) is obtained, w^{k-1} is prolongated to grid k, i.e.,

$$w^k_{\text{new}} = w^k_{\text{old}} + I^k_{k-1} w^{k-1}. \tag{23}$$

The restriction and polongations on grid k are continued until the residuals decrease below a desired accuracy.

A number of variants to the above outline of the multigrid method have been proposed, including extensions to nonlinear problems. For complete review of the MG technique, excellent reviews by Brandt [15], Stuben and Trottenberg [17], and the proceedings of recent multigrid conferences [19, 20] may be consulted.

4. Details of Present Implementation

In this section, we describe the details of the overall algorithm, the smoothing operator and the restriction/prolongation relations employed for the block-implicit solution of the Navier–Stokes equations. Because of the coupling between the equations, a special smoother called SCGS (symmetrical coupled Gauss–Seidel) is developed; it is observed to provide good smoothing. The SCGS is similar to the Box Gauss–Seidel scheme mentioned by Brandt [15].

Overall Multigrid Cycle

We have employed the FAS–FMG (full approximation storage–full multigrid) algorithm originally developed by Brandt [15] and subsequently used by many investigators, including Ghia, Ghia, and Shin [21]. The FAS–FMG, which is ideally suited for nonlinear problems, is a generalization of the correction scheme (CS). The flow chart of present iterative cycle, shown in Fig. 2, proceeds as follows. After a series of grids is chosen, iterations are initiated on the coarsest grid (grid number 1). On this grid, the solution of the complete nonlinear problem is sought. This includes performing Newton iterations and solving the linear equations either directly or iteratively. The converged solution on this grid is then prolongated to the next finest grid and a few relaxation sweeps are made. Since the problem is nonlinear, the coefficients A^u_e, A^u_w, A^u_n, A^u_s, etc., are reevaluated within the relaxation sweep. When the smoothing on this grid gets worse below a threshold rate η, i.e., when

$$R_{p+1}/R_p > \eta, \tag{24}$$

the residuals are restricted to the coarser grid and smoothed on this grid. However, unlike the correction scheme, in the FAS procedure the values calculated on a coarser grid $(k-1)$ are not simple corrections to the values on grid k; instead they are approximations on grid $(k-1)$ to the correct values on grid k. In the FAS procedure, therefore, the equations solved on $(k-1)$ are

$$L^{k-1} w^{k-1} = F^{k-1} + I^{k-1}_k (F^k - L^k w^k). \tag{25}$$

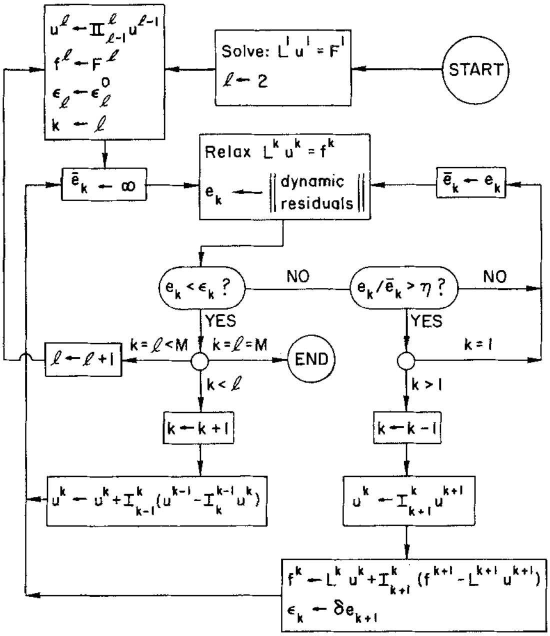

$u^{\ell} \leftarrow I^{\ell}_{\ell-1} u^{\ell-1}$
$f^{\ell} \leftarrow F^{\ell}$
$\epsilon_{\ell} \leftarrow \epsilon^{0}_{\ell}$
$k \leftarrow \ell$
Solve: $L^1 u^1 = F^1$
$\ell \leftarrow 2$
START
Relax $L^k u^k = f^k$
$e_k \leftarrow$ ‖dynamic residuals‖
$\bar{e}_k \leftarrow \infty$
$\bar{e}_k \leftarrow e_k$
$e_k < \epsilon_k$?
NO
$e_k/\bar{e}_k > \eta$?
NO
YES
YES
$k=\ell<M$
$k=\ell=M$
END
$\ell \leftarrow \ell+1$
$k<\ell$
$k=1$
$k>1$
$k \leftarrow k+1$
$k \leftarrow k-1$
$u^k \leftarrow u^k + I^k_{k-1}(u^{k-1} - I^{k-1}_k u^k)$
$u^k \leftarrow I^k_{k+1} u^{k+1}$
$f^k \leftarrow L^k u^k + I^k_{k+1}(f^{k+1} - L^{k+1} u^{k+1})$
$\epsilon_k \leftarrow \delta e_{k+1}$

FIG. 2. Flowchart of FAS–FMG.

Here L is the nonlinear operator consisting of convection and diffusion terms in the equations. The correction to w^k is then

$$w^k_{\text{new}} = w^k_{\text{old}} + I^k_{k-1}(w^{k-1} - I^{k-1}_k w^k_{\text{old}}). \tag{26}$$

Note that in Eq. (26), only the *change* from the previous value $(w^{k-1} - I^{k-1}_k w^k_{\text{old}})$ is prolongated to grid k and not the value w^{k-1} itself. This detail is important in the FAS scheme; otherwise, convergence of overall sequence may not be good. Before the iterations, w^{k-1} is set equal to $I^{k-1}_k w^k_{\text{old}}$.

The iterations on each grid level are continued until the required convergence criterion is met, at which time the solution vector is transferred to the next finer level. When the finest level is solved to the desired accuracy, the overall solution cycle is terminated. Note that the tolerance level on any grid is equal to the originally prescribed value only when that grid is the current finest grid. However, when the current level, k, is less than the current finest level l, the tolerance on k, ε_k is set to

$$\varepsilon_k = \delta e_{k+1}, \tag{27}$$

where e_{k+1} is the norm of error on grid $(k+1)$. Typically, $\delta = 0.2$. As can be seen, the complete FAS–FMG process is nested with many loops over the coarse grids.

The operator on a coarse grid $(k-1)$ during the smoothing of the *restricted* residuals is calculated from the restricted grid function $I^{k-1}_k w^k$. However, as for the fine grid, the operator is linearized around the current solution, w^k. New coefficients A_e^u, A_w^u, etc. are calculated from relations (9)–(16). Thus,

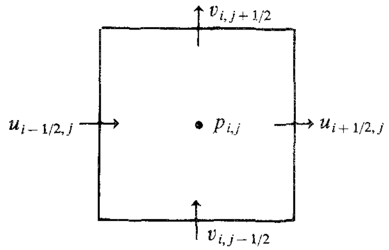
$$L^{k-1} = L(I_k^{k-1} w^k). \tag{28}$$

An alternate way is to evaluate the coefficients A_e^u, A_w^u, etc. of the coarse grid by averaging the corresponding fine grid coefficients. The latter procedure avoids some computations, but in a few trial calculations, it was found to be either slow in convergence or divergent. The first method was therefore used in the present study.

Smoothing Operator

The choice of an efficient smoothing operator (relaxation procedure) is an important aspect in the multigrid technique. Although the concept of the multigrid technique is valid even with poor smoothing operators, the efficiency of the method can be downgraded significantly if the smoothing is not efficient. The choice of the relaxation procedure is somewhat problem dependent and there is always a tradeoff between a robust technique with a larger operation count and a less robust but simpler technique with lower operation count. In the past, a number of smoothers have been used on a variety of problems, including nonlinear equations.

In this study, the momentum and continuity equations are relaxed in a coupled manner. This choice stems from our earlier observations [11–13] with a single grid (using direct solution) that for internal viscous flows, the treatment of the coupling between the pressure and velocity fields is very crucial to procuring rapid convergence of the algorithm (see also [7]). The simultaneous relaxation also follows Brandt's recommendation [15, pp. 33] that "a locally strongly coupled block of unknowns which is locally decoupled (or weakly coupled with) from the coarser-grid variables should be relaxed simultaneously." Ghia *et al.* [21] have used a coupled strongly implicit procedure (CSIP) developed by Rubin and Khosla [22] to simultaneously relax the streamfunction and vorticity equations. The technique used in this study is called the symmetrical coupled Gauss–Seidel (SCGS) procedure. In the SCGS scheme four velocities and one pressure corresponding to one finite-difference node are simultaneously updated by inverting a (small) matrix of equations. Thus velocities on all *four* sides of a cell are updated each time. This contrasts with an unsymmetrical coupled GS (UCGS) in which only one velocity in each direction and the pressure are updated simultaneously. The UCGS which was initially tried was observed to have poor smoothing and often led to divergence. The details of the SCGS scheme are as follows.



Consider the finite-difference node (i, j), shown above. The pressure is located at the center and the velocities are located staggered on four sides. The equations for these five variables are

$$(A_c^u)_{i-1/2,j} u_{i-1/2,j} = F_{i-1/2,j}^u, \tag{29}$$

$$(A_c^u)_{i+1/2,j} u_{i+1/2,j} = F_{i+1/2,j}^u, \tag{30}$$

$$(A_c^v)_{i,j-1/2} v_{i,j-1/2} = F_{i,j-1/2}^v, \tag{31}$$

$$(A_c^v)_{i,j+1/2} v_{i,j+1/2} = F_{i,j+1/2}^v, \tag{32}$$

and

$$(u_{i+1/2,j} - u_{i-1/2,j})/\delta x + (v_{i,j+1/2} - v_{i,j-1/2})/\delta y = 0, \tag{33}$$

where

$$\begin{aligned} F_{i+1/2,j}^u = {} & (A_n^u)_{i+1/2,j} u_{i+1/2,j+1} + (A_s^u)_{i+1/2,j} u_{i+1/2,j-1} \\ & + (A_e^u)_{i+1/2,j} u_{i+3/2,j} + (A_w^u)_{i+1/2,j} u_{i-1/2,j} \\ & + (p_{i,j} - p_{i+1,j})/\rho\, \delta x, \text{ etc.} \end{aligned} \tag{34}$$

In terms of residuals and corrections at node (i, j), the equations can be written as

$$\begin{aligned} (A_c^u)_{i+1/2,j} u'_{i+1/2,j} - p'_{i,j}/\rho\, \delta x = R_{i+1/2,j}^u = F_{i+1/2,j}^u - (A_c^u)_{i+1/2,j} u_{i+1/2,j} \\ (A_c^u)_{i-1/2,j} u'_{i-1/2,j} + p'_{i,j}/\rho\, \delta x = R_{i-1/2,j}^u = F_{i-1/2,j}^u - (A_c^u)_{i-1/2,j} u_{i-1/2,j}, \end{aligned} \tag{35}$$

etc. The above equations can be arranged in a block structure, as follows:

$$\begin{bmatrix} (A_c^u)_{i-1/2,j} & 0 & 0 & 0 & 1/\rho\, \delta x \\ 0 & (A_c^u)_{i+1/2,j} & 0 & 0 & -1/\rho\, \delta x \\ 0 & 0 & (A_c^v)_{i,j-1/2} & 0 & 1/\rho\, \delta y \\ 0 & 0 & 0 & (A_c^v)_{i,j+1/2} & -1/\rho\, \delta y \\ -1/\delta x & 1/\delta x & -1/\delta y & 1/\delta y & 0 \end{bmatrix} \times \begin{bmatrix} u'_{i-1/2,j} \\ u'_{i+1/2,j} \\ v'_{i,j-1/2} \\ v'_{i,j+1/2} \\ p'_{i,j} \end{bmatrix} = \begin{bmatrix} R_{i-1/2,j}^u \\ R_{i+1/2,j}^u \\ R_{i,j-1/2}^v \\ R_{i,j+1/2}^v \\ R_{i,j}^c \end{bmatrix}. \tag{36}$$

The above block of equations are inverted analytically by considering the matrix as a bordered one. Note that the coefficients A_c^u, A_c^v are not differentiated. Because of this approximate Jacobian and the evaluation of the coefficients by successive substitution, it was necessary to underrelax the successive changes in the velocities.

Such underrelaxation is similar to that performed in algorithms such as SIMPLE and its variants [7] that also seek the steady state directly. The under relaxation technique has the effect of a false time dependent term in the equations and has stabilizing characteristics. The underrelaxation was implemented by modifying A_c^u, A_c^v as follows:

$$(A_c^u)_{i-1/2,j} = (A_c^u)_{i-1/2,j}/\alpha_u \tag{37}$$

$$(A_c^v)_{i,j-1/2} = (A_c^v)_{i,j-1/2}/\alpha_v, \text{ etc.} \tag{38}$$

In the present calculations equal values were used for α_u and α_v. Their optimum value, which varied with the flow Reynolds number was between 0.5 and 0.8. Smaller values were necessary for higher Reynolds numbers. It is, however, not necessary to use constant values of underrelaxation factors. These can be varied, for more effectiveness, as suggested by Raithby and Schneider [7] and by Neuberger, Chatwani, Eickhoff and Koopman [31].

The SCGS scheme was motivated by the concept that it has some similarity to the solution of the Poisson equation with the traditional Gauss–Seidel scheme. The SCGS essentially corrects the pressure and the fluxes (velocities) out of a control cell simultaneously. The SCGS contrasts with an unsymmetrical update procedure in which only the forward or backward velocities are modified along with the pressure. In contrast with Brandt's distributive Gauss–Seidel (DGS), the SCGS employes a coupled update, whereas DGS is a decoupled solution procedure, i.e., DGS solves all momentum equations first and then satisfies the continuity equation by correcting velocities and pressures. Because the SCGS scheme satisfies the (linearized) pointwise momentum and continuity equations simultaneously it can be advantageous over DGS in situations where the coefficients A_c^u, A_c^v vary on the sides of the control volume. One such case is when one of the cell faces is partially blocked (by a baffle or flame holder, etc.). The SCGS, however, updates each velocity component essentially twice. The assembling of the coefficients and residuals requires roughly 100 floating point operations (multiply, divide, or add) and the inversion of the bordered matrix and correction of variables require 46 floating point operations.

The choice of a Gauss–Seidel scheme in lieu of several others was purely for reasons of simplicity. Other schemes such as line relaxation on the coupled equations are also possible. The yet even simpler Jacobi iteration is, however, not feasible in the symmetrical update strategy. This is because any velocity is common to two cells and the corrections calculated at the two cells in a Jacobi update manner are not superimposable. An additional disadvantage of Jacobi iteration is the requirement of two sets of arrays, one for the variables and the other for the corrections. In the SCGS scheme, there is no need for any storage for corrections because they are applied immediately.

Restriction and Prolongation

The restriction and prolongation procedures are somewhat dictated by the staggered mesh arrangement. Restriction is used for transferring fine grid values to a coarse grid, whereas prolongation is used for extrapolating a coarse grid correction to a fine grid. The two operators were denoted earlier by I_k^{k-1} and I_{k-1}^k, respectively. A frequent restriction operator is injection, i.e., the coarse grid value is taken to be the local fine grid value. Thus

$$w_{i,j}^c = w_{2i-1,2j-1}^f, \tag{39}$$

where the superscripts c, f denote coarse and fine grid values. The injection operator is not applicable for the staggered mesh because of the different locations of variables on coarse and fine meshes (see Fig. 1). In the current study, the restrictions are made by averaging nearby values. Let (ic, jc) and (if, jf) denote coarse and fine mesh indices, respectively. Also, let $u_{i+1/2,j}$ be referred to as $u_{i,j}$ and $u_{i-1/2,j}$ be referred to as $u_{i-1,j}$, etc. Then $if = 2(ic) - 1$, $jf = 2(jc) - 1$, and

$$u^c(ic, jc) = \tfrac{1}{2}[u^f(if, jf) + u^f(if, jf-1)],$$
$$v^c(ic, jc) = \tfrac{1}{2}[v^f(if, jf) + v^f(if-1, jf)],$$

and

$$p^c(ic, jc) = \tfrac{1}{4}[p^f(if, jf) + p^f(if-1, jf) + p^f(if, jf-1) + p^f(if-1, jf-1)]. \tag{40}$$

The prolongation relations are derived by a bilinear interpolation. For each coarse grid node, four fine grid values are derived. For u-velocity, they are

$$\begin{aligned}
u^f(if, jf) &= \tfrac{1}{4}(3u_1^c + u_2^c),\\
u^f(if, jf+1) &= \tfrac{1}{4}(3u_2^c + u_1^c),\\
u^f(if+1, jf) &= \tfrac{1}{8}(3u_1^c + u_2^c + 3u_3^c + u_4^c),\\
u^f(if+1, jf+1) &= \tfrac{1}{8}(3u_2^c + u_1^c + 3u_4^c + u_3^c),\\
u_1^c = u^c(ic, jc); &\qquad u_2^c = u^c(ic, jc+1);\\
u_3^c = u^c(ic+1, jc); &\qquad u_4^c = u^c(ic+1, jc+1).
\end{aligned} \tag{41}$$

The v-velocities can be prolongated by equivalent relations obtained by rotating the coordinates by ninety degrees. The pressure prolongations have different weightings because of their cell-centered locations. For each coarse grid cell,

$$\begin{aligned}
p^f(if, jf) &= \tfrac{1}{16}(9p_1^c + 3p_2^c + 3p_3^c + p_4^c),\\
p^f(if, jf+1) &= \tfrac{1}{16}(9p_3^c + 3p_4^c + 3p_1^c + p_2^c),\\
p^f(if+1, jf) &= \tfrac{1}{16}(9p_2^c + 3p_1^c + 3p_4^c + p_3^c),
\end{aligned}$$

and

$$p^f(if+1, jf+1) = \tfrac{1}{16}(9p_4^c + 3p_2^c + 3p_3^c + p_1^c), \tag{42}$$

with

$$\begin{aligned} p_1^c &= p^c(ic, jc); & p_2^c &= p^c(ic+1, jc); \\ p_3^c &= p^c(ic, jc+1); & p_4^c &= p^c(ic+1, jc+1). \end{aligned} \tag{43}$$

The above relations are slightly modified near the boundaries to avoid using the boundary pressures. Such relations are obtained by assuming a zero derivative condition near the boundaries.

The prolongation operator to interpolate coarse grid solution to the next finer grid (I^l_{l-1} in Fig. 2) may or may not be the same as the prolongation operator for the corrections. In the present study, for simplicity, the same above operators are used for both corrections and the solution. More accurate cubic prolongation operators for the solution have been considered before [15, 21] and can further accelerate convergence. However, cubic or higher order polynomial fits are somewhat inconvenient to use in practical flows with complex geometrical shapes and blocked regions (although they are not considered in this study).

It is necessary to remember that in the FAS–FMG algorithm, the values on the coarse grid are not directly prolongated; instead the *changes* from the previously restricted values are prolongated. That is,

$$w^k_{\text{new}} = w^k_{\text{old}} + I^k_{k-1}\,\delta w^{k-1} \tag{44}$$

and

$$\delta w^{k-1} = w^{k-1} - I^{k-1}_k w^k_{\text{old}}. \tag{45}$$

Relations (41), (42) are used on δw^{k-1}.

5. Application to Flow in a Driven Cavity

The numerical solution of the flow in a rectangular cavity with the top wall moving at a constant velocity has been a standard problem for testing the efficiency of many solution algorithms. The problem characterizes the elliptic and nonlinear nature of many engineering flows. The flow in a square cavity has also been of interest because of the yet unresolved controversies on the flow structure at high Reynolds numbers ($\geqslant 5000$). A number of numerical studies have been published (e.g., [23, 24, 25]) although the accuracy of many of the results at high Reynolds numbers remains doubtful. Some have employed the upwind differencing, while some others have used the hybrid formulation. A review up to 1978 has been made by Tuann and Olson [26] in which the finite element and finite difference approaches have been compared. Recent studies of this problem have been done by Ghia *et al.* [21], Agarwal [27], Benjamin and Denny [28], and Rubin and Khosla [22]. Ghia *et al.* used the coupled SIP with the multigrid concept and stream function-vorticity formulation. Numerical solutions up to Re = 10,000 were obtained with 257 × 257 grid nodes in modest CPU times. The results of Agarwal and of Ghia *et al.* are in good agreement with very recent studies by Schreiber and Keller [29, 30]. In the latter, a number of novel techniques such as continuation methods have been employed to obtain solutions up to Re = 10,000. The technique used here is significantly different from earlier investigations and has wider applicability to other engineering flow situations, including three-dimensional flows.

In the application of the present algorithm to the square cavity problem, a number of flow Reynolds numbers and several finite-difference grids have been considered. Calculations have been made for Reynolds numbers of 100, 400, 1000, 2000, and 5000 with grids consisting of 41 × 41, 81 × 81, 161 × 161, 321 × 321 finite-difference nodes. The boundary conditions were of the Dirichlet type for velocities, and no boundary conditions were necessary for the pressure. The singularity of the system of equations (because of the incompressibility condition) was resolved by holding the pressure fixed at the corner ($I = 2$, $J = 2$) node. The threshold smoothing rate, η, for switching from fine to coarse grids was fixed at 0.5, for all calculations. The sensitivity of the rate of convergence to values of η has not been assessed in this study but will be investigated in the future. All calculations were initiated from zero velocity and pressure fields without using information from previous solutions at lower Reynolds numbers. The convergence criterion was based on the summed average residual in the three equations. Thus,

$$R_{\text{conv}} = \left[\sum_{i,j} \{(R^u_{i,j})^2 + (R^v_{i,j})^2 + (R^c_{i,j})^2\}/(\text{IMAX*JMAX*3}) \right]^{1/2}, \tag{46}$$

where R^u, R^v, and R^c are residuals in the u, v momentum equations and in the continuity equations. The momentum residuals have been normalized by ρu_w^2 and the mass error is normalized by ρu_w, where u_w is the top wall velocity and ρ is the density. R_{conv} was set to 10^{-3} and when the fine grid residual decreased below this

TABLE I

Optimal Underrelaxation Factors

Re Grid	100	400	1000	2000	5000
41 × 41	0.8	0.8	0.6	0.6	0.6
81 × 81	0.8	0.8	0.6	0.6	0.5
161 × 161	0.8	0.8	0.6	0.6	0.5
321 × 321	0.8	0.8	0.5	0.5	—

TABLE II

CPU Times[a] and Number of Iterations

Grid \ Re	100	400	1000	2000	5000
41 × 41	4.09 12 32	6.09 15 47	12.42 26 95	16.43 32 125	21.0 51 160
81 × 81	16.52 15 33	22.0 12 43	52.0 24 100	88.0 33 169	133.0 60 250
161 × 161	64.0 15 31	71.0 14 35	154.0 19 74	280.0 25 133	853.0 52 403
321 × 321	282.0 16 34	267.0 15 32	644.0 23 78	1080.4 27 131	— — —

[a] IBM 3033 s.

value, the calculations were terminated. On the coarsest mesh R_{conv} was 10^{-4}. The nodes were numbered in a lexicographic order. For all the Reynolds numbers, an optimum underrelaxation factor was first evaluated on coarser grids. Because of the increased nonlinearity at higher Reynolds numbers, it was necessary to use lower values of the relaxation parameter at higher Re. It was empirically observed that the CPU times differed at most by a factor of two between the calculations with the optimum value and with a near-optimum value (of course, for excessively large values of this parameter, divergence also resulted). It was also observed that the

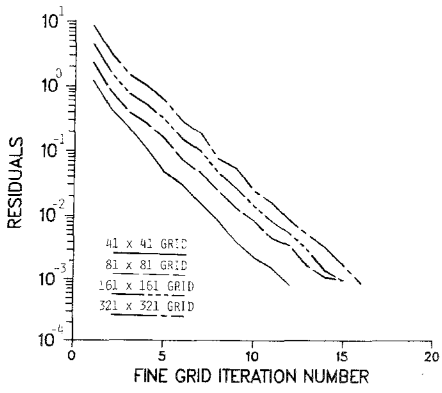


FIG. 3. Rate of convergence for Re = 100.

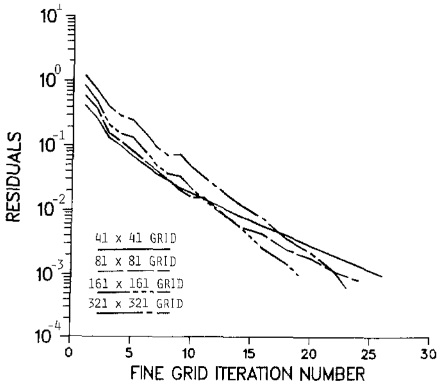


FIG. 4. Rate of convergence for Re = 1000.

optimum underrelaxation factor on the coarse grid (41 × 41) is nearly optimal on finer grids. Table I gives these optimal values for the underrelaxation factors.

The CPU times for the matrix of calculations with the optimal underrelaxation factors are given in Table II. The times are on an IBM3033 machine with FORTHX compiler with OPT(2) optimization. Also given in Table II are numbers of fine-grid iterations and the equivalent work units for the total calculation (a work unit is equal to the time for one fine grid SCGS iteration). From Table II, it can be seen that except for Re = 5000, the CPU time increased almost linearly with the number of grid nodes. This confirms the attractive feature of the multigrid technique. The absolute CPU time, however, increased with Reynolds number, as a result of the increased nonlinearity. Successful calculations were made up to Re = 2000 and grids of 321 × 321 nodes. For the Re = 5000 case, calculations were made only up to the 161 × 161 grid. For this Reynolds number, the calculation with 321 × 321 was very slow in convergence and required heavy underrelaxation. Because it is uncertain whether the flow remains laminar at this high Reynolds

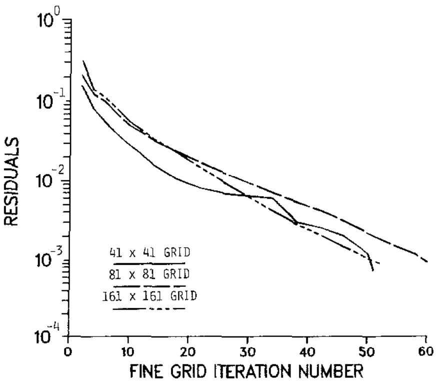


FIG. 5. Rate of convergence for Re = 5000.

TABLE III

Experimentally Observed Smoothing Rates in Initial Iteration with Lexicographic Ordering

Re \ Grid	41 × 41	81 × 81	161 × 161	321 × 321
100	0.36	0.35	0.36	0.36
400	0.45	0.40	0.36	0.36
1000	0.63	0.63	0.57	0.59
2000	0.62	0.66	0.64	0.65
5000	0.67	0.72	0.75	—

number, the calculations with the finest grid were not pursued. Figures 3, 4, 5 show some of the observed rates of convergence.

The rate of smoothing, i.e., proportionate decrease of residuals with iterations is found to vary with iteration number, grid density, and Reynolds numbers. Table III gives some empirically observed smoothing factors in the initial part of the calculations. Also, in in the present study, an adaptive strategy has been used to decide the time of restrictions to coarse grids. Alternatively a fixed cycle multigrid strategy can also be used. It is unknown at this stage which strategy gives the lower work count. Certainly this must be examined in future studies. In comparison with a single grid convergence history, these rates are much faster. For example, for a 41 × 41 grid, Re = 400 calculation, a single grid requires 506 iterations. With the use of multigrid technique, only 47 equivalent iterations are required. For larger grids, the advantage of the multigrid technique is much larger. Figure 6 shows convergence history of three single grid calculations using the SCGS relaxation.

The present CPU times are significantly smaller than the times required by some other primitive variable formulations known to us. However, comparison with all such procedures is a monumental task and even then will not be useful because the

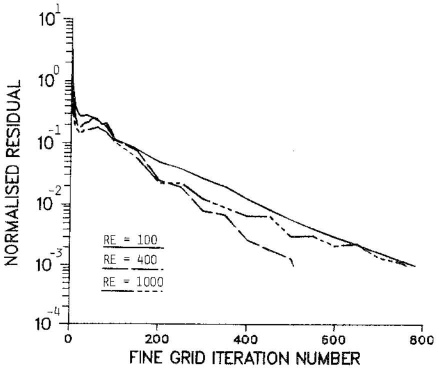


FIG. 6. Rates of convergence for single grid SCGS.

list of competing methods will not be exhaustive. On a very gross level, we have compared present CPU times with those of the SIMPLE algorithm that has been used in many earlier investigations. For Re = 1000 and a grid of 81 × 81 nodes, we observe that the present method is faster by a factor of fifty over SIMPLE. The present times can also be compared with those of Ghia *et al.* [21] using a $\psi \sim \omega$ formulation and coupled SIP multigrid technique. Their times and ours are quite close. However, they terminated their calculations at a lower accuracy level.

In contrast with the direct inversion procedure by LU factorization (e.g., Vanka and Leaf [10]), the use of multigrid techniques requires much less storage. Currently, storage is required only for the variables u, v, and p, and the vectors of residuals. The coefficients are evaluated dynamically during the iteration stage for each node (i, j); hence no storage has been necessary. The variables are stored for all the grids, including the coarse ones. This means that an additional $\frac{1}{3}$ of the storage for the finest grid is required. The residuals, however, are stored only for the coarse grids. The total array storage for a $(m \times m)$ grid is approximately $5m^2$. Thus for the 321 × 321 grid a total job card storage of 550 K words was adequate.

6. Results

The motion of the top wall sets up a complex vortex structure in the cavity. At low Reynolds numbers, the flow consists of a primary vortex and two secondary vortices at the bottom upstream and downstream corners. As the Reyholds number is increased beyond a value of ~1500, another vortex is formed at the upstream (left) top corner of the cavity. Figure 7 shows the contours of stream function for the several Reynolds numbers considered. These flow patterns are in good agreement with earlier results of Schreiber and Keller [30], Agarwal [27], and Ghia *et al.* [21]. The minor vortices embedded in the bottom corners are not plotted here because the contour intervals were selected to be uniform, and the stream function in those locations were of the order of 10^{-5}. Table IV gives the values of

TABLE IV

Selected Characteristics of the Driven Cavity Flow

	Primary vortex			Lower left vortex			Lower right vortex		
Re	$\psi_{\max}$	x	y	$\psi_{\max}$	x	y	$\psi_{\max}$	x	y
100	0.1034	0.6188	0.7375	$-1.94E-6$	0.0375	0.0313	$-1.14E-5$	0.9375	0.0563
400	0.1136	0.5563	0.6000	$-1.46E-5$	0.0500	0.0500	$-6.45E-4$	0.8875	0.1188
1000	0.1173	0.5438	0.5625	$-2.24E-4$	0.075	0.0813	$-1.74E-3$	0.8625	0.1063
2000	0.1116	0.5250	0.5500	$-6.90E-4$	0.0875	0.1063	$-2.60E-3$	0.8375	0.0938
5000	0.0920	0.5125	0.5313	$-1.67E-3$	0.0625	0.1563	$-5.49E-3$	0.8500	0.0813

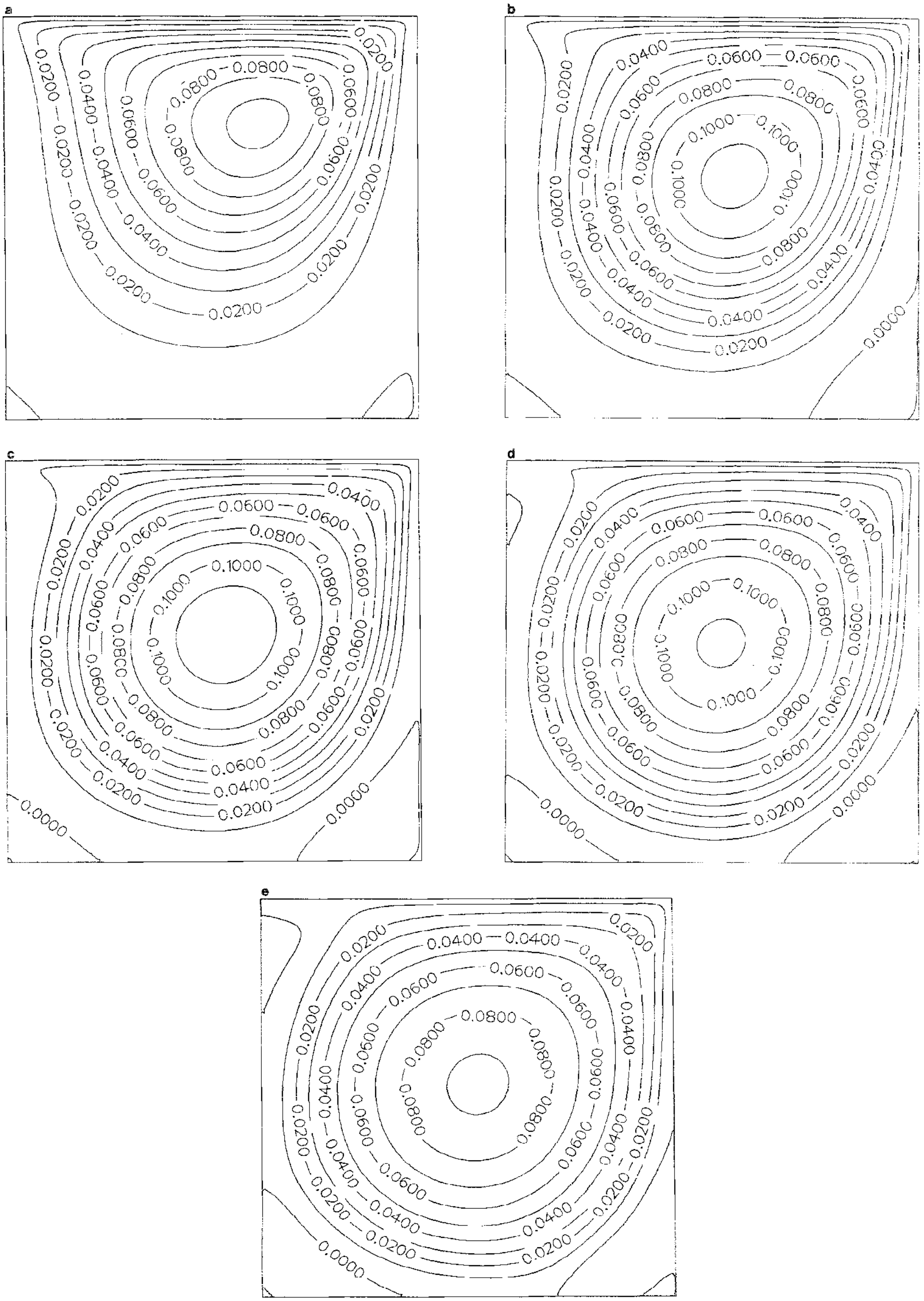


FIG. 7. (a) Contours of streamfunction, Re = 100, 321 × 321 grid. (b) Contours of streamfunction, Re = 400, 321 × 321 grid. (c) Contours of streamfunction, Re = 1000, 321 × 321 grid. (d) Contours of streamfunction, Re = 2000, 321 × 321 grid. (e) Contours of streamfunction, Re – 5000, 161 × 161 grid.

TABLE V

Maximum Negative Velocity (and Location) on Vertical Centerline

Grid Re	41 × 41	81 × 81	161 × 161	321 × 321
100	−0.202 (0.4625)	−0.209 (0.4562)	−0.212 (0.4594)	−0.213 (0.4578)
400	−0.279 (0.2875)	−0.318 (0.2812)	−0.326 (0.2781)	−0.327 (0.2797)
1000	−0.258 (0.1875)	−0.338 (0.1812)	−0.381 (0.1719)	−0.387 (0.1734)
2000	−0.229 (0.1375)	−0.311 (0.1312)	−0.379 (0.1219)	−0.415 (0.1203)
5000	−0.189 (0.0875)	−0.262 (0.09375)	−0.335 (0.08437)	— —

the maximum stream function and the locations in the central, bottom left and bottom right vortices. The values for the central (primary) vortex are in excellent agreement with earlier results. Some differences are found in the bottom corner values; the differences can be justified because the values of the streamfunction are small and, in some cases, below the convergence accuracy of the present as well as the previous calculations.

The effect of the finite-difference grid on the calculated results is shown in Table V by considering the value of the maximum negative velocity on the vertical centerline. It is seen that the 161 × 161 grid is accurate up to Re = 1000 and beyond this value finer grids are necessary. Figure 8 shows a typical velocity profile on the

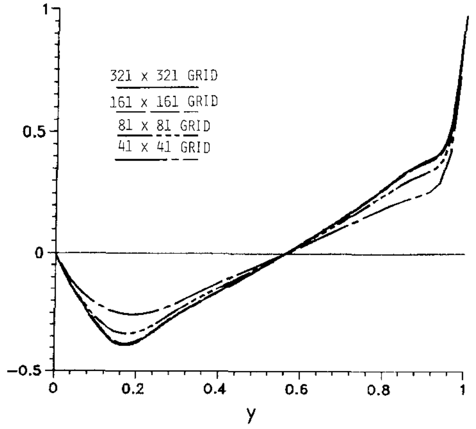


FIG. 8. Velocity profile along vertical centerline, Re = 1000.0.

vertical centerline for Re = 1000. For this Reynolds number, the results for 321×321 grid and the 161×161 grid are in close agreement. These profiles are also in good agreement with earlier results, such as those of Agarwal [27].

7. Summary

An efficient calculation procedure for steady state Navier–Stokes equations in the primitive variable formulation has been developed using a coupled solution of equations by the multigrid technique. A smoothing technique called SCGS has been developed that is observed to provide good smoothing rates. The algorithm is applied to study the flow structure in a square cavity with a moving top wall. Calculations at very high Reynolds numbers have been made in modest computer times. Good convergence has been observed in almost all cases except with very fine grids and very large Reynolds numbers. The computer times are observed to vary as $O(n)$ (except at Re = 5000), which is the theoretically expected rate for the multigrid technique. The current CPU times are significantly smaller than other primitive variable formulations using decoupled solutions and one grid iteration concepts. The current times are comparable with a $\psi \sim \omega$ multigrid procedure, but the (u, v, p) formulation has more applicability to complex geometries and to 3-dimensional flows.

The present flow fields are compared with previously published numerical results and good agreement has been observed. As future work, we shall be extending the SCGS scheme to three dimensions and to turbulent and reacting flows where the fluid density and viscosity vary spatially. Also the efficiency of a coupled ILU factorization as a smoother for highly nonlinear problems is being investigated. Modifications for adaptation on a vector processor are also being considered.

Acknowledgments

I have greatly benefited from helpful discussions with Gary Leaf of Argonne National Laboratory and with Chris Thompson of AERE, Harwell, U.K. (currently visiting Argonne National Laboratory). This work was supported by the Ramjet Technology Division, Wright Patterson AFB under interagency agreement with Argonne National Laboratory. I am grateful to Drs. F. D. Stull and R. R. Craig for their support and encouragement.

References

1. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8**, 2182 (1965).
2. F. H. Harlow and A. A. Amsden, *J. Comput. Phys.* **8**, 197 (1971).
3. C. W. Hirt, B. D. Nichols, and N. C. Romero, Los Alamos Scientific Laboratory Report LA-5852, 1975.
4. S. V. Patankar and D. B. Spalding, *Int. J. Heat Mass Transfer* **15**, 1787 (1972).
5. R. Srinivasan *et al.*, NASA-CR-168423, 1983.

6. G. J. Sturgess, NASA-CR-168202, 1983.
7. G. D. Raithby and G. E. Schneider, *Numer. Heat Transfer* **2**, 417 (1979).
8. S. V. Patankar, *Numer. Heat Transfer* **4**, 409 (1981).
9. R. Issa, Report FS/82/15, Imperial College, London, 1982.
10. G. E. Schneider and M. Zedan, AIAA-84-1743, in *Proceedings of the AIAA/SAE/ASME 20th Joint Propulsion Conference, 1984.*
11. S. P. Vanka and G. K. Leaf, AIAA-84-1244, in *Proceedings of the AIAA/SAE/ASME 20th Joint Propulsion Conference, June 1984.*
12. S. P. Vanka, *Int. J. Heat Mass Transfer* **28**, 2093 (1985).
13. S. P. Vanka, AIAA-85-0141, in *Proceedings of the 23rd Aerospace Sciences Meeting, Reno, Nevada, 1985.*
14. H. L. Stone, *SIAM J. Numer. Anal.* **5**, 530 (1968).
15. A. Brandt, von Karman Institute, Lecture Series 1984-04, 1984.
16. A. Brandt, J. E. Dendy, Jr., and H. Ruppel, *J. Comput. Phys.* **34**, 348 (1980).
17. K. Stüben and U. Trottenberg, *Multigrid Methods*, Lecture Notes in Mathematics Vol. 960, Springer-Verlag, Berlin, 1982.
18. D. B. Spalding, *Int. J. Numer. Methods* **4**, 551 (1972).
19. W. Hackbusch and U. Trottenberg (Eds.), *Multigrid Methods Conference Proceedings, Köln-Porz, 1981.*
20. *Proceedings, 1983 International Multigrid Conference, Copper Mountain, Colorado, April 1983.*
21. U. Ghia, K. N. Ghia, and C. T. Shin, *J. Comput. Phys.* **48**, 387 (1982).
22. S. G. Rubin and P. K. Khosla, *Computers and Fluids* **9**, 163 (1981).
23. M. Nallasamy and K. Krishna Prasad, *J. Fluid Mech.* **79**, No. 2, 391 (1977).
24. J. D. Bozeman and C. Dalton, *J. Comput. Phys.* **12**, 348 (1973).
25. D. R. Burggraf, *J. Fluid Mech.* **24**, 113 (1966).
26. S. Y. Tuann and M. D. Olson, *J. Comput. Phys.* **29**, 1 (1978).
27. R. K. Agarwal, 19th Aerospace Sciences Meeting, St Louis, MO, 1981; MDRL-80-51.
28. A. S. Benjamin and V. E. Denny, *J. Comput. Phys.* **33**, 340 (1979).
29. R. S. Schreiber and H. B. Keller, *J. Comput. Phys.* **49**, 165 (1983).
30. R. Schreiber and H. B. Keller, *J. Comput. Phys.* **49**, 310 (1983).
31. A. W. Neuberger, A. U. Chatwani, H. Eickhoff, and J. Koopman, *Proceedings, Third International Conference on Numerical Methods in Laminar and Turbulent Flow, Seattle, August 1983.*