

NASA Technical Memorandum 102359  
ICOMP-89-24

# Application of Multi-grid Methods for Solving the Navier-Stokes Equations

A.O. Demuren  
*Institute for Computational Mechanics in Propulsion  
Lewis Research Center  
Cleveland, Ohio*

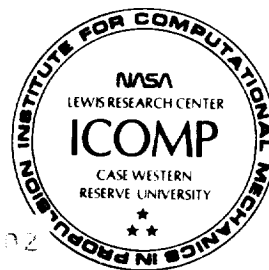
*and University of Lagos  
Lagos, Nigeria*

October 1989



(NASA-TM-102359) APPLICATION OF MULTI-GRID  
METHODS FOR SOLVING THE NAVIER-STOKES  
EQUATIONS (NASA) 23 p CSCL 12A

N90-1400Z



Unclass

63/64 0240386



# APPLICATION OF MULTI-GRID METHODS FOR SOLVING THE NAVIER-STOKES EQUATIONS\*

A.O. Demuren†  
Institute for Computational Mechanics in Propulsion  
Lewis Research Center  
Cleveland, Ohio 44135

and

Faculty of Engineering  
University of Lagos  
Lagos, Nigeria

## SUMMARY

This paper presents the application of a class of multi-grid methods to the solution of the Navier-Stokes equations for two-dimensional laminar flow problems. The methods consist of combining the full approximation scheme - full multi-grid technique (FAS-FMG) with point-, line-, or plane-relaxation routines for solving the Navier-Stokes equations in primitive variables. The performance of the multi-grid methods is compared to that of several single-grid methods. The results show that much faster convergence can be procured through the use of the multi-grid approach than through the various suggestions for improving single-grid methods. The importance of the choice of relaxation scheme for the multi-grid method is illustrated.

E-4655

## INTRODUCTION

In spite of the rapid development of computer hardware over the last two decades, there is need for continuous improvement of computational techniques in order to achieve better accuracy with minimal computational effort. In most numerical methods, for fluid flow calculations, better accuracy is obtained through the use of higher-order discretizations, or finer grid distributions, or a combination of both. With a large number of grid points, direct methods of solution are often not feasible, because of memory limitations, so an iterative method must be used. However, it has been found that increasing the number of grid points distributed in the computational domain leads to a deterioration of the convergence rate. Thus, for most practical problems in two- or three-dimensions, the grid distribution required for accurate solution would lead to excessive computational times (of the order of hours for two-dimensional and days for three-dimensional calculations), even on very large mainframe computers. Hence, the quest for more efficient solution procedures. The multi-grid technique is emerging as a very promising tool for accelerating the convergence rate of iterative procedures, especially for calculations with very fine distributions.

---

\*This paper is reproduced from the Proceedings of the Institution of Mechanical Engineers, Part C, 1989, by permission of the Council of the Institution.

†Work funded under Space Act Agreement C99066G.

The technique for accelerating convergence of an iterative procedure through the use of multiple grids was applied as early as 1935 by Southwell (ref. 1). Other works along similar lines were reported by Southwell (refs. 2 and 3), Stiefel (ref. 4), Federenko (ref. 5) and Wachspress (ref. 6), among others. These were all two-grid level methods. The idea of utilizing fully multiple-grid levels was introduced by Federenko (ref. 7) for a Poisson-type problem on a rectangular grid, and the approach was generalized by Bakhlov (ref. 8) to any second-order elliptic operator with continuous coefficients. The first actual numerical computations with the full multi-grid technique were reported by Brandt (ref. 9) for some boundary-value problems, using a finite-difference approach. The multi-grid method has been applied with finite-element formulations by Nicolaides (ref. 10). In most of these, linear differential equations were solved. More recently, multi-grid techniques for solving nonlinear equations have been reported by Brandt (ref. 11), Ghia et al. (ref. 12) and Vanka (ref. 13).

The previous studies show that the multi-grid methods are quite promising but a lot of work still needs to be carried out in order to realize their full potential. In the present study, the multi-grid technique is applied, with some finite-difference methods, to solve the Navier-Stokes equations for two-dimensional model flow problems. Their performance is compared with that of various single-grid methods for varying grid distributions and Reynolds numbers.

## MATHEMATICAL MODEL

### The Multi-grid Concept

Consider a differential problem represented over a given domain  $D$  by the linear equation:

$$L U(\bar{x}) = F(\bar{x}) \quad (1)$$

where  $L$  is the differential operator,  $U$  is the unknown variable,  $F$  is a known function, and  $\bar{x} = (x_1, x_2, \dots, x_d)$  are the  $d$  independent variables of the  $d$ -dimensional problem. If the domain is divided into a computational grid with spacing  $h$ , the finite-difference equivalent of the equations may be written as:

$$L_h U_h(\bar{x}h) = F_h(\bar{x}h) \quad (2)$$

Solution of this equation by an iterative procedure such as Gauss-Seidel, Jacobi, line relaxation, etc., have been found to converge rapidly only for the first few iterations, and more slowly thereafter. By considering a Fourier analysis of the error-reduction process of typical relaxation procedures, Brandt (ref. 11) showed that they are only efficient in smoothing out those error components whose wavelengths are comparable to the grid mesh size. Error components with longer wavelengths are smoothed out at comparatively slower rates. Thus, the idea of the multi-grid technique is to smooth out high frequency error components by performing a few iterations of the relaxation process on the fine grid. The remaining errors are then transferred to a coarser grid where the corresponding lower frequency error components are again smoothed out in a few iterations. Further transfers are made to even coarser

grids and the process is repeated until all the error components have been smoothed out. The results are then progressively transmitted back to the finer grids. The overall effect of this procedure is that the various Fourier components of the error are removed on grid meshes most efficient for the purpose, thereby accelerating the convergence rate on the fine grid on which the solution to the finite difference equation is sought.

Thus, in the multi-grid method, the computation is carried out on a series of grid meshes  $D^k$  with the corresponding grid functions  $U^k$ , where  $k = 1, 2, 3, \dots, M$ , with  $k = M$  representing the finest mesh, and the meshes becoming coarser the lower the value of  $k$ . The mesh sizes usually differ by a factor of 2, so that  $h_{k+1} = 1/2 h_k$ . Although this is not a requirement of the multi-grid method, it enables simple weighting functions to be employed in interpolation routines. The finite difference equation can then be written for the  $k^{\text{th}}$  grid as:

$$L^k U^k = F^k \quad (3)$$

There are several algorithms for implementing the multi-grid idea, each with several possible variations. One of the simplest is the correction scheme (CS) which is applicable to linear problems. In this scheme, the calculation starts on the finest grid  $D^M$ , and an approximate solution to equation (3) is computed by a relaxation method such as the Gauss-Seidel, line relaxation, etc. Unless the approximate solution  $U^k$  (fortuitously) satisfies the difference equation (3) and the boundary conditions, there will be a residual  $R^k$  given by:

$$F^k - L^k U^k = R^k \quad (4)$$

A few iterations are performed on grid  $D^M$  until the rate of reduction of the residuals falls below a desired theoretical rate. The residuals are then transferred by restriction to the next coarser grid,  $D^{M-1}$  and a correction function  $\delta U^{k-1}$  is obtained by solving the equation:

$$L^{k-1} \delta U^{k-1} = I_k^{k-1} R^k \quad (5)$$

where  $L^{k-1}$  is the difference operator on grid  $D^{k-1}$  and  $I_k^{k-1}$  is the restriction operator. If  $k-1$  is the coarsest grid, equation (5) is solved completely on this grid, otherwise, the problem is transferred to the next coarser grid as soon as the rate of smoothing of the residuals falls below the required norm. Once equation (5) has been solved, the correction function  $\delta U^{k-1}$  is prolonged to grid  $D^k$ , and  $U^k$  is subsequently corrected as:

$$U_{\text{new}}^k = U_{\text{old}}^k + I_{k-1}^k \delta U^{k-1} \quad (6)$$

where  $I_{k-1}^k$  is the prolongation operator. Similar equations are employed to transfer (prolongate) the intermediate corrections from the coarser grids, once the prescribed convergence criteria have been satisfied. This process of relaxation, restriction and prolongation is repeated until the desired accuracy on the finest grid,  $D^M$  is achieved.

The scheme described above serves to illustrate the multi-grid idea, but is inadequate for nonlinear problems such as the Navier-Stokes equation, of interest here. For this, the full approximation storage (FAS) scheme is employed.

In the FAS scheme, the full approximation  $U^{k-1}$  is stored on a coarse grid rather than its correction  $\delta U^{k-1}$  as in the CS scheme.  $U^{k-1}$  is given by:

$$U^{k-1} = I_k^{k-1} U^k + \delta U^{k-1} \quad (7)$$

where  $U^k$  is the fine grid approximation. The full approximation can be obtained on the coarse grid by solving the corresponding FAS equation:

$$L^{k-1} U^{k-1} = L^{k-1} (I_k^{k-1} U^k) + I_k^{k-1} (F^k - L^k U^k) \quad (8)$$

The fine grid approximation is then corrected as:

$$U_{\text{new}}^k = U_{\text{old}}^k + I_{k-1}^k (U^{k-1} - I_k^{k-1} U_{\text{old}}^k) \quad (9)$$

Comparison of equations (5) and (8) shows that the CS and FAS schemes are equivalent for linear problems. However, equation (8) is additionally suitable for nonlinear problems, for which the former is not valid.

#### Present Implementation

In the present paper, the FAS-FMG (full approximation storage - full multi-grid) algorithm originally developed by Brandt (ref. 11) is employed. The procedure commences by solving iteratively until convergence, the finite difference equations on the coarsest grid, using a relaxation routine. The solution is then prolonged to the next finer grid, where it serves as the initial approximation of the solution to the difference equations on this grid. Improved solutions are obtained by applying the relaxation routine on this grid, until the smoothing rate falls below a desired value. The optimum value was found by numerical experiments to be a rate of reduction of the normalized average residual of 0.8 (i.e.,  $R_{n+1}/R_n \leq 0.8$ ). Otherwise, the task of smoothing the residuals is transferred to a coarser grid and the FAS scheme already described is employed. This rate happens to correspond to the theoretical smoothing rate computed by Brandt (ref. 11) for the Navier-Stokes equation. The iteration on each grid level is continued until the prescribed convergence criterion is satisfied, at which time, the solution is transferred to the next finer grid level and the whole process repeated. The solution process is terminated on satisfying this criterion at the finest grid level  $DM$ .

It is only necessary to iterate until full convergence on the finest and coarsest grids. Otherwise, the finer grid residuals are considered sufficiently smoothed once the corresponding coarse grid errors have been reduced to about 20 percent of the restricted values. Then the corrections are prolonged back to the fine grid. In the present work, fully adaptive, full multi-grid cycles are employed, which are controlled solely by the aforementioned convergence criteria and the desired smoothing rate. There is, however, a

wide range of possible multi-grid cycles, some of which are discussed by Stüben and Trottenberg (ref. 14).

### Restriction and Prolongation

Restriction and prolongation routines are required to transfer the fine grid approximations and residuals onto the coarse grid, and the coarse grid corrections onto the fine grid, respectively. The operators were denoted by  $I_k^{k-1}$  and  $I_{k-1}^k$ , respectively. In the present work, a staggered grid is employed for the velocity components and the pressure, as shown in figure 1. So, the transfer operators would be different for each of the variables U,V,P and for the residuals  $R_U$  and  $R_V$ . The restrictions are made by averaging the neighboring fine grid values, and the prolongations by applying bilinear interpolation to the closest set of coarse grid values.

### Relaxation Methods

The Navier-Stokes equations can be written for two-dimensional steady, laminar plane flow as:

$$\frac{\partial(UU)}{\partial x} + \frac{\partial(UV)}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \left( \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \right) \quad (10)$$

$$\frac{\partial(UV)}{\partial x} + \frac{\partial(VV)}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \left( \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} \right) \quad (11)$$

and the continuity equation as:

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \quad (12)$$

Using the staggered grid system (fig. 1), these equations can be discretized for the  $i,j$ <sup>th</sup> cell giving, in each case:

$$A_c^U U_{i,j} = A_e^U U_{i+1,j} + A_w^U U_{i-1,j} + A_n^U U_{i,j+1} + A_s^U U_{i,j-1} + \frac{P_{i-1,j} - P_{i,j}}{\rho \Delta x_{i-1/2,j}} \quad (13)$$

$$A_c^V V_{i,j} = A_e^V V_{i+1,j} + A_w^V V_{i-1,j} + A_n^V V_{i,j+1} + A_s^V V_{i,j-1} + \frac{P_{i,j-1} - P_{i,j}}{\rho \Delta y_{i,j-1/2}} \quad (14)$$

$$\frac{U_{i+1,j} - U_{i,j}}{\Delta x_{i,j}} + \frac{V_{i,j+1} - V_{i,j}}{\Delta y_{i,j}} = 0 \quad (15)$$

where the coefficients  $A_c, A_e, A_n$ , etc., are derived using the hybrid (central/upwind) difference scheme proposed by Spalding (ref. 15). The actual expressions can be found in Vanka (ref. 13).

Similar expressions can be written for the other face velocities by increasing the indices of  $i$  and  $j$  by one in equations (13) and (14), respectively. Thus:

$$A_c^u U_{i+1,j} = A_e^u U_{i+2,j} + A_w^u U_{i,j} + A_n^u U_{i+1,j+1} + A_s^u U_{i+1,j-1} + \frac{P_{i,j} - P_{i+1,j}}{\rho \Delta x_{i+1/2,j}} \quad (16)$$

$$A_c^v V_{i,j+1} = A_e^v V_{i+1,j+1} + A_w^v V_{i-1,j+1} + A_n^v V_{i,j+2} + A_s^v V_{i,j} + \frac{P_{i,j} - P_{i,j+1}}{\rho \Delta y_{i,j+1/2}} \quad (17)$$

We now assume that the velocities on the right-hand side of equations (13), (14), (16), and (17) are known, so that the face velocities (on the left-hand side) can be evaluated once the pressures are found. The method for finding the pressures characterizes the relaxation scheme. In equations (13) to (17), there are four unknown velocities and five unknown pressures. Each of these can be split into known and unknown components, consisting of guesses (\*)'s, and corrections (')'s. Thus,

$$\left. \begin{aligned} U_{i,j} &= U_{i,j}^* + U'_{i,j} \\ U_{i+1,j} &= U_{i+1,j}^* + U'_{i+1,j} \\ V_{i,j} &= V_{i,j}^* + V'_{i,j} \\ V_{i,j+1} &= V_{i,j+1}^* + V'_{i,j+1} \end{aligned} \right\} \quad (18)$$

and

$$P_{i,j} = P_{i,j}^* + P'_{i,j} \quad (19a)$$

$$P_{i+1,j} = P_{i+1,j}^* + P'_{i+1,j} \quad (19b)$$

$$P_{i-1,j} = P_{i-1,j}^* + P'_{i-1,j} \quad (19c)$$

$$P_{i,j+1} = P_{i,j+1}^* + P'_{i,j+1} \quad (19d)$$

$$P_{i,j-1} = P_{i,j-1}^* + P'_{i,j-1} \quad (19e)$$

We now need to solve the system of equations (13) to (17) for the corrections, i.e., the (')'s

#### Plane Relaxation (SIM)

In equations (13), (14), (16), and (17), each of the velocity corrections is expressed solely in terms of known velocities and pressures and unknown pressure corrections, so that we can substitute these equations into equation (15) to yield the pressure correction equation. This equation will have five unknowns, and it will be necessary to generate the equations at all the internal nodes in the two-dimensional plane in order to form a closed set.



The penta-diagonal matrix equation set has the form:

$$E_{i,j}P'_{i,j} = A_{i,j}P'_{i+1,j} + B_{i,j}P'_{i-1,j} + C_{i,j}P'_{i,j+1} + D_{i,j}P'_{i,j-1} + F_{i,j} \quad (20)$$

where

$$A_{i,j} = 1 / \left( \rho A_{C_{i+1,j}}^U \Delta x_{i,j} \Delta x_{i+(1/2),j} \right)$$

$$B_{i,j} = 1 / \left( \rho A_{C_{i,j}}^U \Delta x_{i,j} \Delta x_{i-(1/2),j} \right)$$

$$C_{i,j} = 1 / \left( \rho A_{C_{i,j+1}}^V \Delta y_{i,j} \Delta y_{i,j+(1/2)} \right)$$

$$D_{i,j} = 1 / \left( \rho A_{C_{i,j}}^V \Delta y_{i,j} \Delta y_{i,j-(1/2)} \right)$$

$$E_{i,j} = A_{i,j} + B_{i,j} + C_{i,j} + D_{i,j}$$

and

$$F_{i,j} = R_{i,j}^C + 1 / \Delta x_{i,j} \left( R_{i,j}^U / A_{C_{i,j}}^U - R_{i+1,j}^U / A_{C_{i+1,j}}^U \right) \\ + 1 / \Delta y_{i,j} \left( R_{i,j}^V / A_{C_{i,j}}^V - R_{i,j+1}^V / A_{C_{i,j+1}}^V \right)$$

The R's represent the residuals of equations (13) to (17). The equation set must be solved simultaneously for all internal grid nodes in the two-dimensional plane. The strongly implicit method (SIM) of Stone (ref. 16) is applied with an iteration parameter of 0.9 for the solution. Once all the pressure corrections are obtained, the velocity corrections can be computed from:

$$U'_{i,j} = \left[ R_{i,j}^U + (P'_{i-1,j} - P'_{i,j}) / \rho \Delta x_{i-(1/2),j} \right] / A_{C_{i,j}}^U \quad (21)$$

$$U'_{i+1,j} = \left[ R_{i+1,j}^U + (P'_{i,j} - P'_{i+1,j}) / \rho \Delta x_{i+(1/2),j} \right] / A_{C_{i+1,j}}^U \quad (22)$$

$$V'_{i,j} = \left[ R_{i,j}^V + (P'_{i,j-1} - P'_{i,j}) / \rho \Delta y_{i,j-(1/2)} \right] / A_{C_{i,j}}^V \quad (23)$$

$$V'_{i,j+1} = \left[ R_{i,j+1}^V + (P'_{i,j} - P'_{i,j+1}) / \rho \Delta y_{i,j+(1/2)} \right] / A_{C_{i,j+1}}^V \quad (24)$$

Due to the nonlinearity of the momentum equations it may be necessary to under-relax the velocity corrections so that only a fraction of them is utilized in equation (18). The usual practice of implicit under-relaxation is employed here. The optimum under-relaxation factor was found to be 0.9.

### Line Relaxation (CLSOR)

Line relaxation is obtained, using a coupled, line successive over-relaxation scheme. If we presume that only the pressure corrections for nodes lying along a line (in the x- or y-direction) are unknown, all others are known, presumably zero. Then there will be three unknown pressure corrections and four unknown velocity corrections to be determined at each node. Equation (20) then reduces, for x-line relaxation, to one of the form:

$$E_{i,j} P'_{i,j} = A_{i,j} P'_{i+1,j} + B_{i,j} P'_{i-1,j} + F_{i,j} \quad (25)$$

Similar equations need to be written for the other internal nodes along the line, in order to form a closed set. The system of tri-diagonal equations is then solved simultaneously, with a tri-diagonal matrix algorithm (TDMA), which is efficient and has a low operation count. The velocity corrections are subsequently computed, for all the nodes lying along the line, using a variant of equations (21) to (24) in which the pressure corrections  $P'_{i,j-1}$  and  $P'_{i,j+1}$  have been set equal to zero. The procedure is then repeated for the other parallel lines in the computational domain. Y-line relaxation can be performed in a similar manner. The optimal under-relaxation factor was found to be 0.6 for test case 1 and 0.9 for test case 2.

### Point Relaxation

This is the so-called symmetric coupled Gauss Seidel (SCGS) method proposed by Vanka (ref. 13). It assumes that only the pressure correction at the central node is unknown so that there are five equations and five unknowns (one pressure correction and four velocity corrections) to be determined at each node. These equations can be solved algebraically, since simple manipulations yield an explicit expression for the pressure correction. The velocity corrections are then computed from a variant of equations (21) to (24) in which all pressure corrections other than the central one are set to zero. The procedure is repeated for all internal grid nodes. A lexicographic visiting order, which is reversed in alternate iterations, was employed in the present study. The optimum under-relaxation factor was found to be between 0.5 and 0.6 for this method. The SCGS differs from the SIVA (simultaneous variable adjustment) method of Caretto et al. (ref. 17) only through the incorporation of additional unknown velocities on the right-hand side in the latter. For example, on the right-hand side of equations (13) and (14), respectively,  $U_{i+1,j}$  and  $V_{i,j+1}$ , are also considered unknown. Similarly, in equations (16) and (17)  $U_{i,j}$  and  $V_{i,j}$  are considered unknown. The explicit expression for the central pressure correction is more complex in SIVA.

### Comparison of Relaxation Methods

In the present relaxation methods, the pressure-velocity coupling, inherent in the momentum and continuity equations, is retained. Further, all the

velocity components are obtained simultaneously. They differ from the SIMPLE method of Patankar and Spalding (ref. 18), and proposed derivatives (refs. 19 and 20) which solve for the velocity components in a sequential or decoupled manner.

The plane-relaxation method is the most implicit, and would normally be expected to be the most stable, having also the fastest convergence rate. However, the coefficients of the equations require two-dimensional storage arrays, so that the computer memory requirements are high. On the other hand, the point-relaxation method is the least implicit, exhibiting a space decoupling from neighboring grid points. The decoupling effect is reduced somewhat by computing, at each node, all the four velocities on the faces of the control volume. Hence, in a complete sweep of the computational domain, each velocity is updated twice. However, no arrays are required to store any coefficient of the equations. The line-relaxation method lies in between. Only the velocities normal to the relaxation line are computed twice. Also, the coefficients only need be stored along one line of grid nodes. The increase in memory requirements is thus minimal.

The expected superiority of plane relaxation is not fully realized in the present study, since the methods are applied to nonlinear problems. The equations are linearized and the coefficients have to be repeatedly updated as soon as better values are available. The point-relaxation method ensures the most rapid updating of coefficients, and the plane-relaxation method, the slowest. Hence, the advantages of increased implicitness in the latter are offset, somewhat, by slower updating of coefficients, and vice versa.

## RESULTS AND DISCUSSION

### Test Cases

Three test cases were chosen in order to evaluate the performance of the present methods. The first is the laminar flow in a lid-driven square cavity (see fig. 2(a)). The second case is the laminar flow through a backward facing step with an expansion ratio of 2 (see fig. 2(b)). Finally, the lid-driven cavity flow was computed in several cavities over a range of aspect ratios.

For case 1, computations were performed at two Reynolds numbers, namely 100 and 1000, based on the velocity of the moving lid and the cavity width. For case 2, computations were performed at Reynolds numbers (based on the mean inlet velocity and channel height) of 100 and 500 and aspect ratios  $L/H$  of 3 and 6, respectively. The longer aspect ratio of the computational domain was due to the correspondingly longer recirculation length encountered in the latter.

The final case compares the performance of the multi-grid methods in lid-driven cavities with aspect ratios ( $B/H$ ) ranging from  $1/6$  to 6, each at a Reynolds number of 100. The aspect ratio was varied by changing  $H$ , while keeping all other parameters unchanged.

In all these test cases, the same number of grid nodes was utilized in both the  $x$ - and  $y$ -directions, so that the grid cell aspect ratio would be the same as the aspect ratio of the channel or cavity.

## Comparison of Single-Grid and Multi-grid Methods

In order to evaluate the performance of the present multi-grid methods, their convergence characteristics are compared with those of various single-grid methods. Three additional single-grid methods are employed, namely, the SIMPLE algorithm of Patankar and Spalding (ref. 18), an improved variant by Issa (ref. 19) named PISO, and another by Van Doormaal and Raithby (ref. 20) called SIMPLEC. Full details of these methods can be obtained from the original references and will not be repeated here. It suffices to mention that they are all based on the line-by-line solution of the Navier-Stokes equations in a decoupled manner. Thus, they differ from the present line-relaxation method which solves the Navier-Stokes equations, symmetrically along a line, and in a coupled manner.

Table I shows the comparison of the CPU times and the number of iterations required for convergence when the six single-grid methods and three multi-grid methods were applied to the lid-driven square cavity flow. The results are presented for six different grid levels. The convergence criterion in each case is that the residual norm  $\bar{R} < 10^{-5}$ .  $\bar{R}$  is defined as:

$$\bar{R} = \sqrt{\frac{R_U^2 + R_V^2 + R_m^2}{3 \times (NI-2) \times (NJ-2)}}$$

where

$R_U$       sum of residuals of the U-momentum equation at all nodes

$R_V$       sum of residual of the V-momentum equation at all nodes

$R_m$       sum of residual mass sources at all nodes

(NI-2)    number of internal grid nodes in the x-direction

(NJ-2)    number of internal grid nodes in the y-direction

At this convergence level all methods gave the same results. In the columns for the multi-grid methods, three values are presented. The first value represents CPU times on the Siemens 7881 Computer. Although some of the computations were performed on the Definicon DSI/780 processor, test computations showed that these take 25 times longer, so the times have been duly converted. The second value represents the number of iterations on the finest grid and the third gives the total work units which is the equivalent number of finest grid iterations of the total relaxation work performed on all the levels. The latter is thus comparable to the number of iterations in the single-grid columns.

The results show that among the single-grid methods which solve the equations in a coupled manner, plane relaxation produced convergence in the least number of iterations, followed by line relaxation, and then point relaxation.

However, it required the largest CPU time per iteration, hence, total CPU times are comparable for all methods. Contrary to expectations, the decoupled single-grid methods generally resulted in faster convergence than the coupled ones. This is probably due to the faster rate of updating the coefficients in the former, which appears to outweigh the disadvantages of solving the equations sequentially. This may not apply to the multi-grid methods, since in the decoupled procedure, the residuals, which have to be smoothed on the coarser grids, are changed by the pressure correction equation in an inconsistent manner.

Arakawa et al. (ref. 21) reported slight advantages for the coupled, multi-grid methods over the decoupled ones. In general, the multi-grid methods converged faster than single-grid ones, especially for finer grids. Figures 3 and 4 present log-log plots of the CPU times for convergence versus the total number of internal grid points. The slopes roughly approach 2 for single-grid methods, but are approximately 1 for multi-grid methods, which shows that the CPU times for convergence increases almost quadratically with the number of grid points in the latter, but only linearly in the former. It is this independence of the computational work per grid point, on the level of grid refinement that makes multi-grid methods quite attractive for fine-grid calculations. Usually, less than 30 iterations were required on the finest grid. However, in contrast to the single-grid methods, the multi-grid methods showed slower convergence with increase in Reynolds number, in the lid-driven square cavity flow. Typically, 1.5 to 2.5 times more computational work was required at  $Re = 1000$ , as compared with  $Re = 100$ . This coincides with the findings of Vanka (ref. 13) and Barcus et al. (ref. 22).

At higher Reynolds numbers, the coarser grids appear to become inefficient in smoothing out finer grid errors. The point-relaxation method produced the fastest convergence of all multi-grid methods. With the additional advantage of minimal memory requirements to store the coefficients of the governing equations as compared to the other methods, it is clearly the best for this problem. For the finest grid ( $162 \times 162$ ), the ratio of CPU times (best single-grid/best multi-grid) was 81 at Reynolds number of 100, but only 14 at Reynolds number of 1000. Computer core memory requirements were 1.50, 1.52, and 3.30 M Bytes for the multi-grid codes with point, line, and plane relaxations, respectively.

Figure 5 shows some comparisons of the computed velocity profiles with those of Ghia et al. (ref. 12) at Reynolds numbers of 100 and 1000. The agreement is seen to be perfect, within the limits of accuracy of the plots. The predicted sizes and strengths of the primary, secondary, and tertiary vortices also compare quite well. Figure 6 shows the streamline plots at both Reynolds numbers.

Table II shows the comparison of the CPU times and the number of iterations required for convergence for the backward-facing step flow problem. The comparisons are for 6 grid levels. The results show that, for single grids, the point-relaxation method exhibited the slowest convergence rate, and this became worse, the higher the grid aspect ratio. The coupled line- and plane-relaxation schemes had better convergence rates; albeit worse than the better decoupled methods. The deterioration of the convergence rate of the point-relaxation scheme with increased grid aspect ratio was even more pronounced in the multi-grid version. Typically, there was a four- to six-fold increase in

the CPU times for convergence when the grid aspect ratio was increased from 3 to 6. Larger aspect ratios are required for higher Reynolds number computations since the recirculation zone and the recovery length increased with Reynolds number. Preliminary calculations showed that there was no Reynolds number dependence of convergence rates in this case. The multi-grid methods using line- and plane-relaxation schemes showed only modest increases in the required CPU times. The line-relaxation scheme appeared to have the best overall performance. Brandt (ref. 11) did point out that point-relaxation schemes would not be efficient for equations with large differences in magnitudes of the directional coefficients, as would result from grids with high cell aspect ratio. The asymptotic convergence rates of point-relaxation schemes such as the Gauss-Seidel were shown to be inferior to those of line-relaxation schemes. The present results show that the use of the appropriate multi-grid method enabled the required CPU times, on the finest grid, to be reduced by factors of up to about 20 and 13 for  $L/H = 3$  and 6, respectively, as compared with the best single-grid method. The factors are correspondingly lower for the coarser grids. In fact, significant advantages of the multi-grid methods only occurred for grids finer than  $34 \times 34$ . Below this, the best multi-grid method might converge slower than the best single-grid one. It should be noted, however, that the multi-grid concept always resulted in faster convergence, if the same relaxation scheme was utilized. Also, the linearity of the CPU times for convergence with the total number of grid nodes is again shown in the log-log plots in figures 7 and 8. A useful measure of the utility of the multi-grid technique is the ratio of the CPU times for best multi-grid method/ best single-grid method. Table III presents this ratio for the two test cases.

Finally, the three multi-grid methods were applied to the lid-driven cavity flow at Reynolds number of 100, but at various aspect ratios ( $B/H$ ). This test case combines aspects of the first two test cases; the flow reversal in the cavity with high cell aspect ratios. The computed results are similar to those of Pan and Acrivos (ref. 23), with a single central vortex at the low aspect ratios and multiple central vortices at higher ratios of 3 and 6. The convergence characteristics are presented in table IV. The convergence rates for the three calculation methods are similar for aspect ratio of 1. On decreasing the aspect ratio to  $1/6$ , through  $1/3$  all methods experienced much slower convergence rates. CPU times for convergence increased by factors of 7, 34, and 18, for the calculation methods with plane-, line-, and point-relaxation schemes, respectively. Similarly, on increasing the aspect ratio to 6, CPU times for convergence increased by factors of 6 and 7 for the plane- and point-relaxation schemes, respectively. No convergence solution was obtained with the line-relaxation scheme. The residual norm decreased by two orders of magnitude and, then subsequently, oscillated about this value. Line relaxation is, of course, directionally sensitive. It is most effective along lines perpendicular to the main flow direction, sweeping from upstream to downstream. In the present flow problems, the relaxation line would be mostly parallel to the velocity vectors when the aspect ratio is  $1/6$ , but perpendicular to them when the aspect ratio is 6. So the slowing down of the convergence rate in the former is explicable. But in the latter, the velocity vectors have reversed directions near the lid and on the opposite side of the cavity. Whereas, line relaxation is carried out from upstream to downstream with respect to the flow in the top half, the situation is reversed with respect to the flow in the bottom half. This latter situation is believed to be the cause of the oscillation and lack of convergence in this case. The directional sensitivity of line-relaxation schemes make them unsuitable for use in general purpose codes.

Point- and plane-relaxation schemes do not have such directional sensitivity, but the former is more adversely affected at all aspect ratios differing from unity. Plane-relaxation schemes thus appear to be the safest compromise for a general-purpose multi-grid code. But there is a penalty to pay for increased computer core memory requirements. Point-relaxation schemes would, of course, be best if large cell aspect ratios can be avoided.

## CONCLUSION

It has been shown that considerable improvements in convergence rates can be achieved in two-dimensional fluid flow computations on fine grids by employing the multi-grid concept. For nearly homogeneous difference equations, the multi-grid method with a point-relaxation scheme proposed by Vanka (ref. 13) gives the best performance in terms of both CPU times and computer memory requirements. However, this method is inadequate for solving inhomogeneous equations, resulting from large grid aspect ratios, or similar. In these cases, a line- or plane-relaxation scheme should be utilized with the multi-grid method, but the computer memory requirements would increase accordingly. In the particular cases presented here, plane relaxation appeared to be the most robust although it would not necessarily procure the fastest convergence in all problems.

The multi-grid methods showed a linear increase in CPU times for convergence, as the computational grid was increased, whereas the single-grid methods exhibited a nearly quadratic increase. Thus, for very fine-grid computations multi-grid methods would always be better than any single-grid method.

## REFERENCES

1. Southwell, R.V.: Stress-Calculation in Frameworks by the Method of 'Systematic Relaxation of Constraints'. Proc. R. Soc. London, vol. 151A, no. 872, Aug. 1, 1935, pp. 56-95.
2. Southwell, R.V.: Relaxation Methods in Engineering Science. Oxford University Press, New York, 1940.
3. Southwell, R.V.: Relaxation Methods in Theoretical Physics. Clarendon Press, Oxford, 1946.
4. Stiefel, E.L.: Uber einige Methoden der Relaxationsrechnung. Z.A.M.P., vol. 3, 1952, pp. 1-133.
5. Federenko, R.P.: A Relaxation Method for Solving Elliptic Difference Equations. Z. Vycisl, Mat. Mat. Fiz., vol. 1, 1961, pp. 922-927.
6. Wachspress, Z.L.: Iterative Solution of Elliptic Systems, and Applications to the Neutron Diffusion Equations of Reactor Physics. Prentice Hall, 1966.
7. Federenko, R.P.: On the Speed of Convergence of an Iteration Process. Z. Vycisl, Mat. Mat. Fiz., vol. 4, 1964, pp. 559-564.

8. Bakhvalov, N.S.: On the Convergence of a Relaxation Method with Natural Constraints on an Elliptic Operator. *Z. Vycisl, Mat. Mat. Fiz.*, vol. 6, 1966, pp. 861-885.
9. Brandt, A.: "Multi-level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems. International Conference on Numerical Methods in Fluid Mechanics. 3rd., vol. 1, H. Cabannes and R. Temam, eds., Springer-Verlag, 1973, pp. 82-89.
10. Nicolaides, R.A.: On Multiple Grid and Related Techniques for Solving Discrete Elliptic Systems. *J. Comput. Phys.*, vol. 19, no. 4, Dec. 1975, pp. 418-431.
11. Brandt, A.: Multi-level Adaptive Solutions to Boundary-Value Problems. *Math. Comput.*, vol. 31, no. 138, Apr. 1977, pp. 330-390.
12. Ghia, U.; Ghia, K.S.; and Shin, C.T.: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *J. Comput. Phys.*, vol. 48, no. 3, Dec. 1982, pp. 387-411.
13. Vanka, S.P.: Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables. *J. Comput. Phys.*, vol. 65, no. 1, July 1986, pp. 138-158.
14. Stuben, K.; and Trottenberg, U.: Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. *Multigrid Methods*, (Lecture Notes in Mathematics, Vol. 960), W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, 1982, pp. 1-176.
15. Spalding, D.B.: A Novel Finite-Difference Formulation for Differential Expressions Involving Both First and Second Derivatives. *Int. J. Numer. Methods Eng.*, vol. 4, July-Aug. 1972, pp. 551-559.
16. Stone, H.L.: Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations. *SIAM J. Numer. Anal.*, vol. 5, no. 3, Sept. 1968, pp. 530-558.
17. Caretto, L.S.; Curr, R.M.; and Spalding, D.V.: Two Numerical Methods for Three-Dimensional Boundary Layer. *Comput. Methods Appl. Mech. Eng.*, vol. 1, no. 1, June 1972, pp. 39-57.
18. Patankar, S.V.; and Spalding, D.B.: A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows. *Int. J. Heat Mass Transfer*, vol. 15, Oct. 1972, pp. 1787-1806.
19. Issa, R.I.: Solution of Implicitly Discretized Fluid Flow Equations by Operator-Splitting. *J. Comput. Phys.*, vol. 62, 1985, pp. 40-65.
20. van Doormaal, J.P.; and Raithby, B.D.: Enhancements of the Simple Method for Predicting Incompressible Fluid Flows. *Numer. Heat Transfer*, vol. 7, Apr.-June 1984, pp. 147-163.



21. Arakawa, CH. et al.: Application of Multigrid Methods for the Coupled and Decoupled Solution of the Incompressible Navier-Stokes Equations. GAMM-Conference on Numerical Methods in Fluid Mechanics, 7th, Proceedings, R.M. DeVille, ed., Friedr. Vieweg und Sohn, Brunswick, Federal Republic of Germany, 1988, pp. 1-8.
22. Barcus, M.; Peric, M.; and Scheurer, G.: A Control-Volume Based Full-Multigrid Procedure for the Prediction of Two-Dimensional, Laminar, Incompressible Flows. GAMM-Conf. on Numerical Methods in Fluid Mechanics, The Proceedings, M. Deville, ed., Friedo Vieweg und Sohn, Brunswick, Federal Republic of Germany, 1988, pp. 9-16.
23. Pan, F.; and Acrivos A.: Steady Flow in Rectangular Cavities. J. Fluid Mech., vol. 28, pt. 4, June 22, 1967, pp. 643-655.

TABLE I. - CPU TIMES AND NUMBER OF ITERATIONS FOR LAMINAR FLOW IN A LID-DRIVEN SQUARE CAVITY

(a) Re = 100

Grid (NI x NJ)	Single-grid <sup>a</sup>						Multi-grid <sup>b</sup>		
	SIMPLE	SIMPLEC	PISO	SCGS	CLSOR	SIM	SCGS + FMG	CLSOR + FMG	SIM + FMG
7 x 7	0.4/29	0.5/42	0.4/18	0.3/27	0.1/23	0.4/17	-----	-----	-----
12 x 12	1.4/55	1.1/38	1.1/26	1.9/91	1.3/62	2.0/54	0.8/19/35	1.0/30/43	2.8/16/69
22 x 22	13.5/162	5.5/61	5.9/47	23.8/324	16.4/194	16/183	3.9/19/43	4.1/20/44	7.5/14/75
42 x 42	184/548	65/187	70/42	254/853	219/641	255/617	13/14/34	15/17/38	16/15/38
82 x 82	2559/1848	885/625	1010/472	3240/2786	2931/2128	3600/1755	38/12/27	55/15/34	59/14/32
162 x 162	N/A <sup>c</sup>	11 800/2000	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	145/11/23	219/15/34	245/11/41

(b) Re = 1000

7 x 7	0.5/37	0.6/53	0.4/21	0.4/34	0.2/29	0.5/22	-----	-----	-----
12 x 12	1.3/49	1.2/45	1.2/29	1.5/68	1.4/65	1.5/43	0.9/36/40	1.4/36/57	4.6/25/110
22 x 22	8.3/97	5.3/59	4.7/36	14.3/192	14.3/168	10/108	4.6/20/49	11.2/42/13	16/32/157
42 x 42	81/244	38/111	34/69	206/702	145/428	107/269	26/23/68	42/33/105	44/28/129
82 x 82	950/687	348/248	373/182	3045/2538	1499/1086	1492/728	102/18/66	131/22/82	153/22/103
162 x 162	N/A <sup>c</sup>	4496/750	4957/549	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	325/13/52	315/12/47	342/16/63

<sup>a</sup>CPU time (sec) on S7881/number of iterations.

<sup>b</sup>CPU time (sec) on S7881/number of fine grid iterations/total work units.

<sup>c</sup>Not available due to excessive CPU time.

TABLE II. - CPU TIMES AND NUMBER OF ITERATIONS FOR LAMINAR FLOW OVER A BACKWARD FACING STEP

(a) Re = 100, L/H = 3

Grid (NI x NJ)	Single-grid <sup>a</sup>						Multi-grid <sup>b</sup>		
	SIMPLE	SIMPLEC	PISO	SCGS	CLSOR	SIM	SCGS + FMG	CLSOR + FMG	SIM + FMG
6 x 6	0.4/35	0.5/51	0.4/13	0.1/36	0.2/15	0.1/11	-----	-----	-----
10 x 10	0.8/39	0.9/46	0.7/19	1.5/113	0.5/37	0.3/27	0.8/31/55	0.4/20/26	0.4/28/34
18 x 18	5.0/88	2.5/42	2.8/32	15/305	6.5/122	4.5/93	3.8/28/65	2.0/23/24	2.3/24/43
34 x 34	61/278	26/120	32/99	154/775	91/418	74/296	16/22/68	11/27/46	12/23/56
66 x 66	812/922	342/376	340/312	N/A <sup>c</sup>	900/1031	760/1020	63/22/65	55/30/55	56/25/62
130 x 130	N/A <sup>c</sup>	4400/1180	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	214/21/56	243/29/58	257/26/67

(b) Re = 500, L/H = 6

6 x 6	0.4/35	0.5/52	0.4/18	0.4/115	0.6/16	0.1/11	-----	-----	-----
10 x 10	1.0/48	1.0/51	0.8/24	4.2/314	0.7/30	0.4/33	2.8/75/181	0.4/24/29	0.5/29/420
18 x 18	5.0/82	3.6/59	3.4/39	50/1022	4.0/68	3.8/788	17/78/274	2.6/27/43	3.1/33/657
34 x 34	74/338	28/123	40/123	364/1846	50/229	94/354	87/55/345	18/37/725	33/66/169
66 x 66	1000/1100	371/404	502/387	N/A <sup>c</sup>	900/1031	964/1329	358/51/365	70/27/70	95/22/118
130 x 130	N/A <sup>c</sup>	4490/1200	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	N/A <sup>c</sup>	1200/41/292	339/34/82	318/25/90

<sup>a</sup>CPU time (sec) on S7881/number of iterations.

<sup>b</sup>CPU time (sec) on S7881/number of fine grid iterations/total work units.

<sup>c</sup>Not available due to excessive CPU time.

TABLE III. - RATIOS OF CPU TIMES  
[Best single-grid/best multi-grid.]

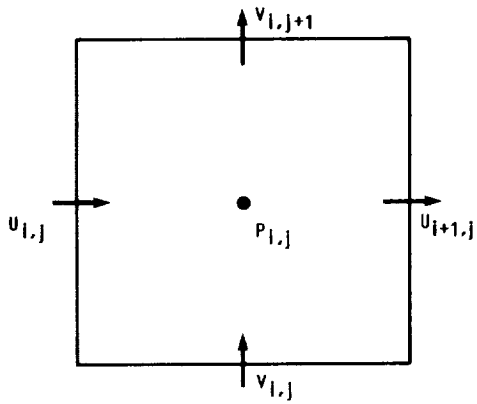
Grid (NI x NJ)	Case 1(a)	Case 1(b)
12 x 12	1.4	1.3
22 x 22	1.4	1.0
42 x 42	5.0	1.3
82 x 82	23.3	3.4
162 x 162	81.4	13.8

Grid (NI x NJ)	Case 2(a)	Case 2(b)
10 x 10	1.3	1.8
18 x 18	1.3	1.3
34 x 34	2.4	1.6
66 x 66	6.2	5.3
130 x 130	20.6	13.2

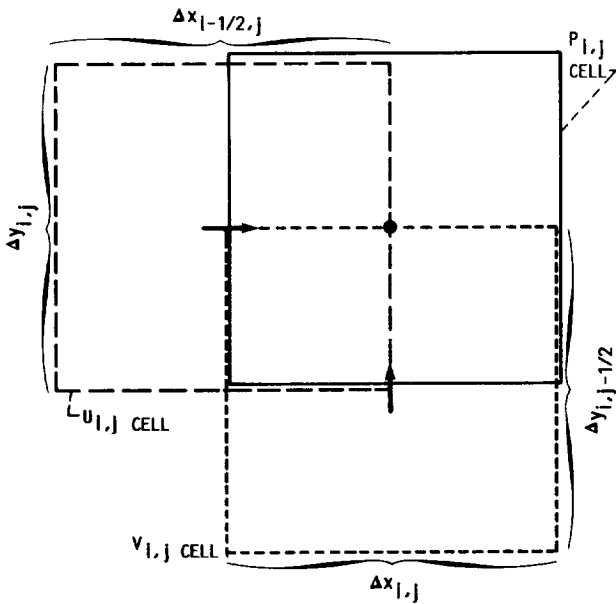
TABLE IV. - PERFORMANCE OF MULTI-GRID METHODS IN LID-DRIVEN FLOW IN CAVITIES WITH DIVERSE ASPECT RATIOS

Aspect ratio (B/H)	Re = 100, 5-level multi-grid, 66 x 66 grid		
	SCGS + FMG <sup>a</sup>	CLSOR + FMG <sup>a</sup>	SIM + FMG <sup>a</sup>
1/6	513/91/484	953/150/1088	237/122/276
1/3	129/35/122	152/ 58/175	95/ 47/115
1	29/11/37	29/ 16/36	32/ 13/27
3	56/27/54	144/ 68/181	64/ 34/65
6	189/97/179	Oscillating, no convergence	184/111/205

<sup>a</sup>CPU times (sec) on S7881/number of fine grid iterations/total work units.

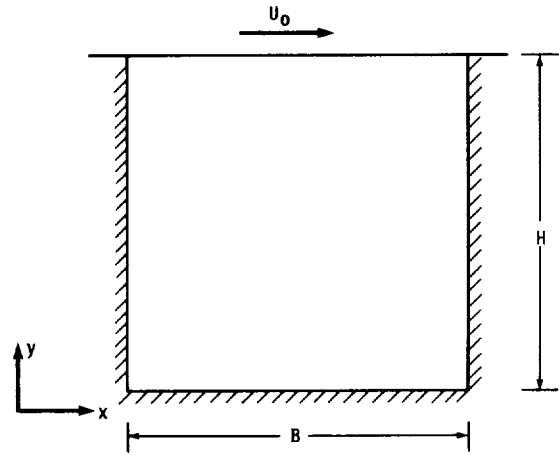


(a) SYMMETRIC GRID NODES.

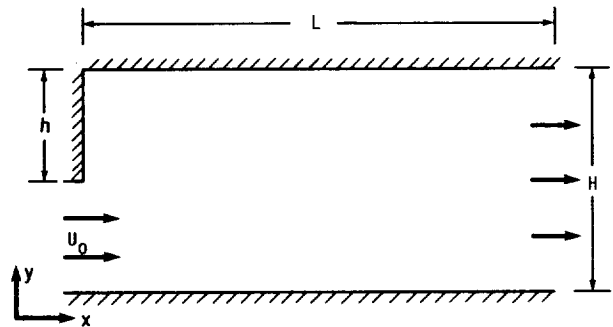


(b) CONTROL VOLUMES FOR THE  $i, j$ th CELL.

FIGURE 1. - STAGGERED GRID SYSTEM.



(a) LID-DRIVEN CAVITY FLOW.



(b) FLOW OVER A BACKWARD-FACING STEP.

FIGURE 2. - ILLUSTRATION OF TEST CASES.

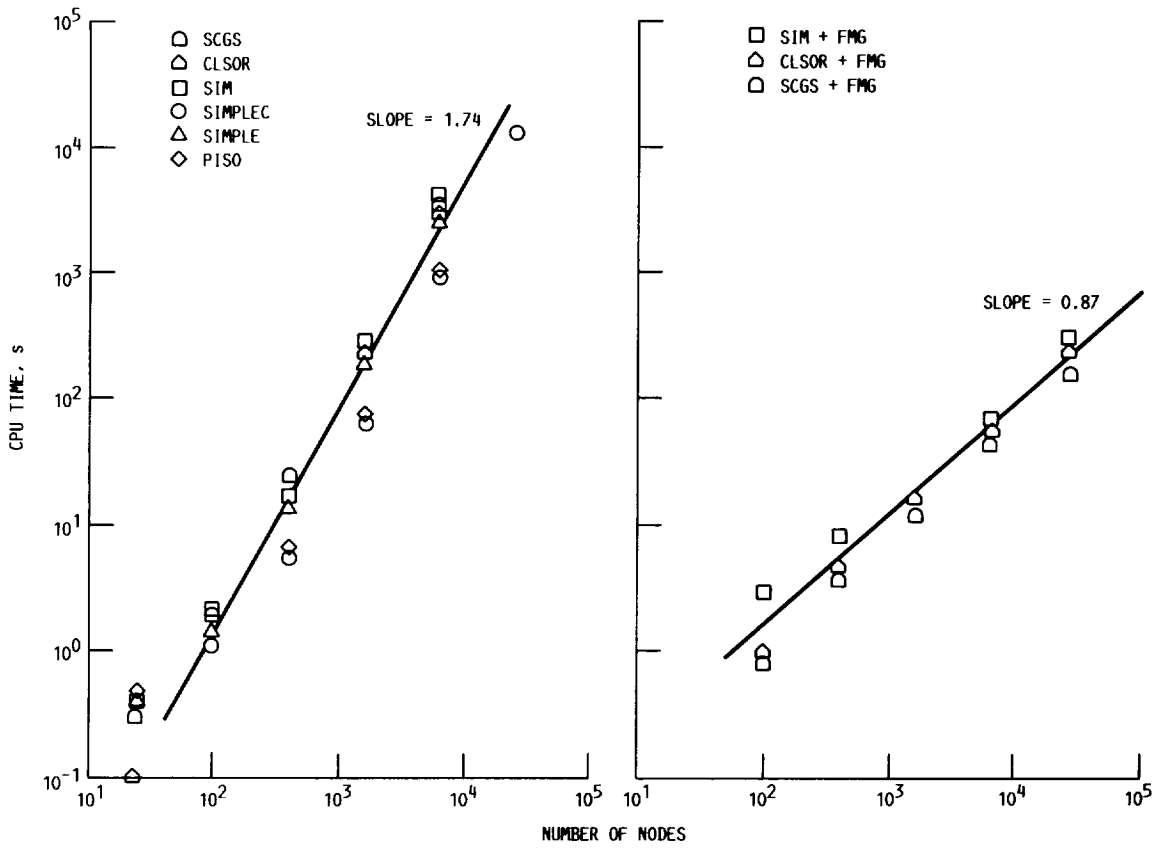


FIGURE 3. - COMPARISON OF CPU TIMES FOR CONVERGENCE; LID-DRIVEN CAVITY FLOW,  $Re = 100$ .

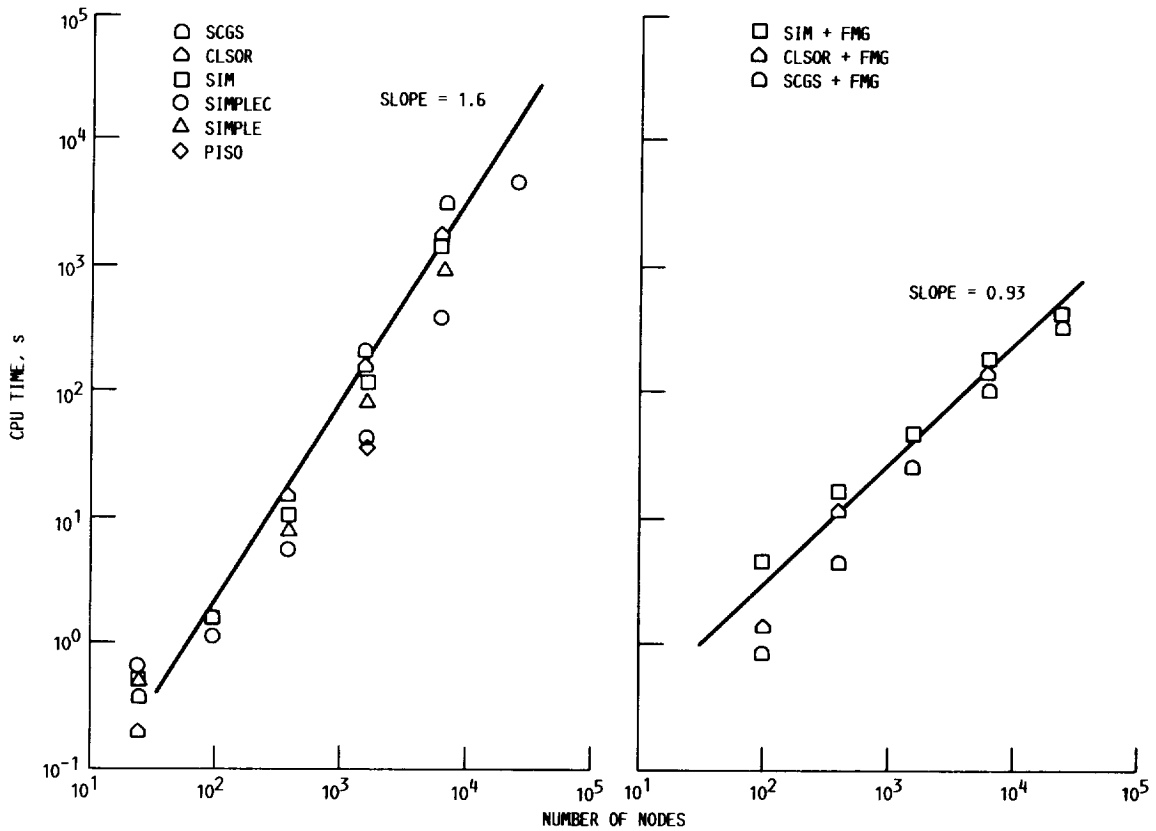
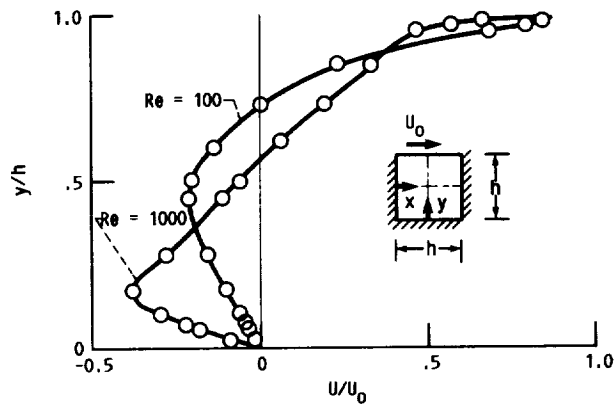
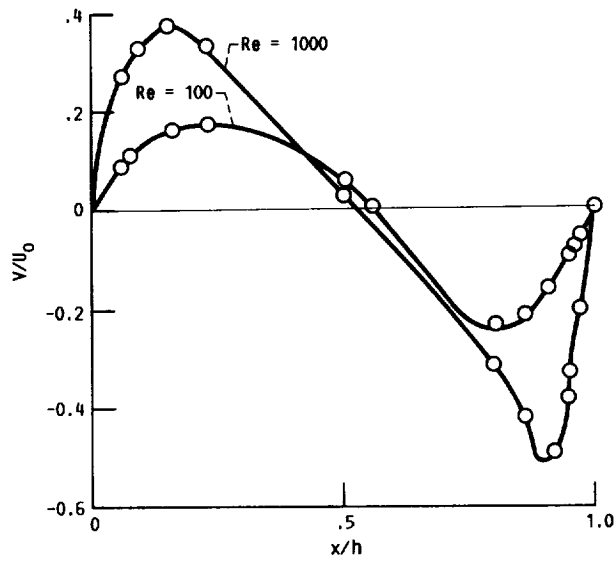


FIGURE 4. - COMPARISON OF CPU TIMES FOR CONVERGENCE; LID-DRIVEN CAVITY FLOW,  $Re = 1000$ .

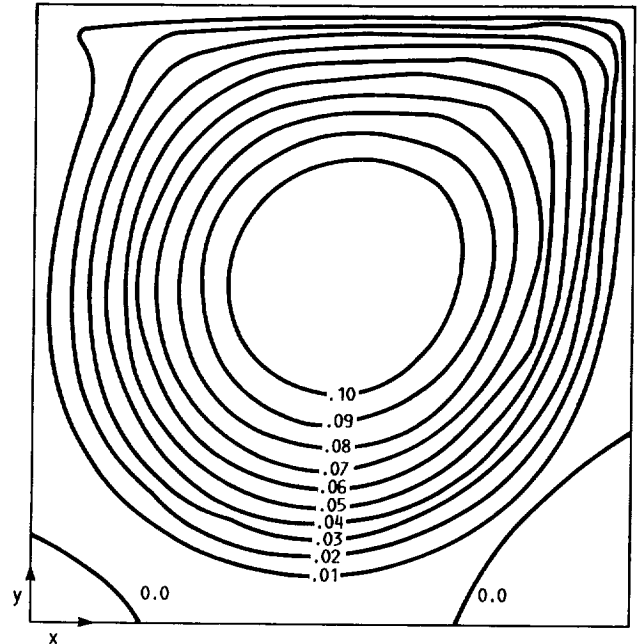


(a) U-VELOCITY ALONG VERTICAL CENTRE LINE.

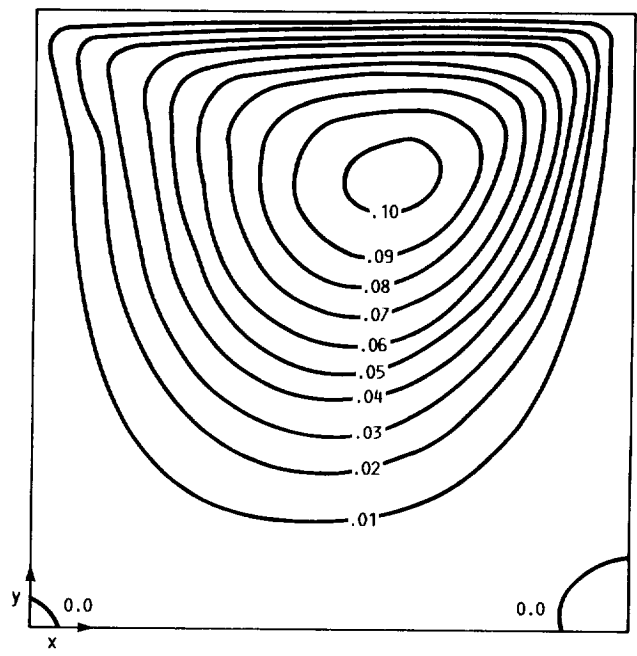


(b) V-VELOCITY ALONG HORIZONTAL CENTRE LINE.

FIGURE 5. - COMPARISON OF VELOCITY PROFILES: PRESENT CALCULATIONS, OOO CALCULATIONS OF GHIA ET AL. (12).



(a)  $Re = 1000.$



(b)  $Re = 100.$

FIGURE 6. - STREAMLINE PLOTS FOR LID-DRIVEN CAVITY FLOWS SHOWING NORMALIZED STREAM FUNCTION ( $\psi/u_0h$ ) CONTOURS.

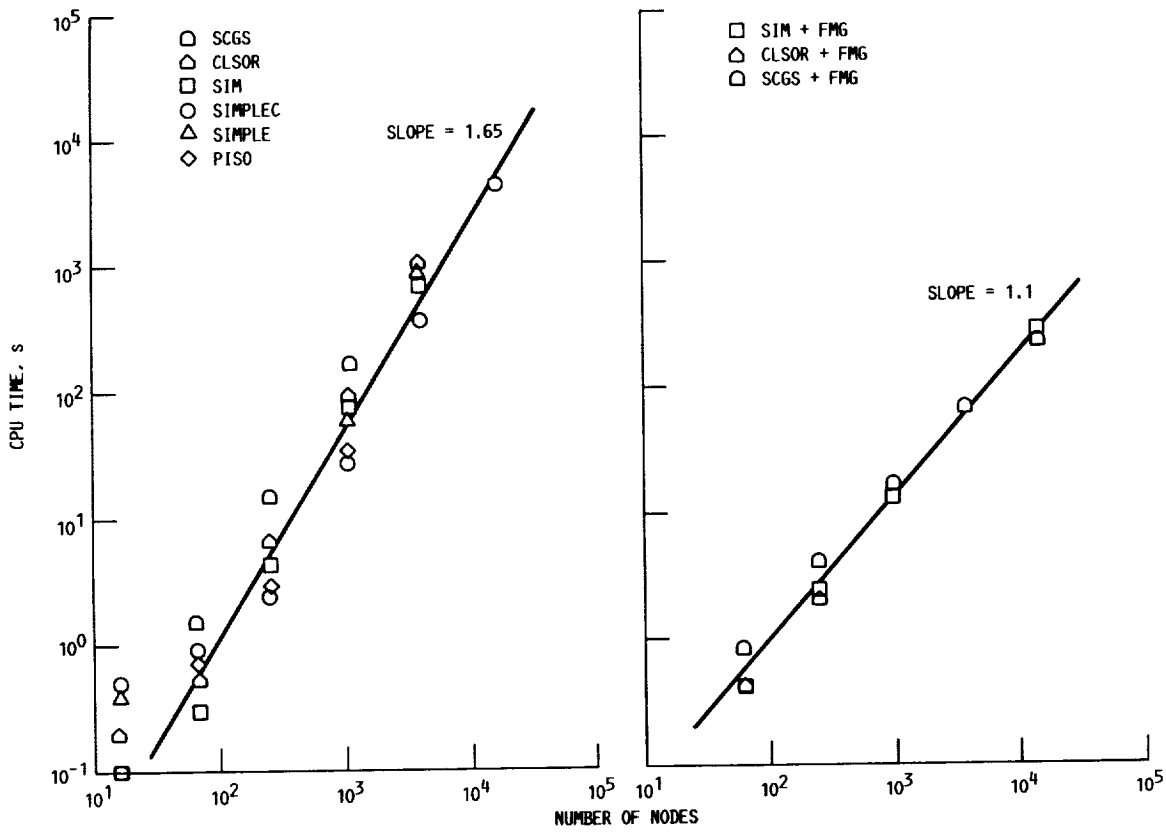


FIGURE 7. - COMPARISON OF CPU TIMES FOR CONVERGENCE; BACKWARD FACING STEP FLOW,  $Re = 100$ .

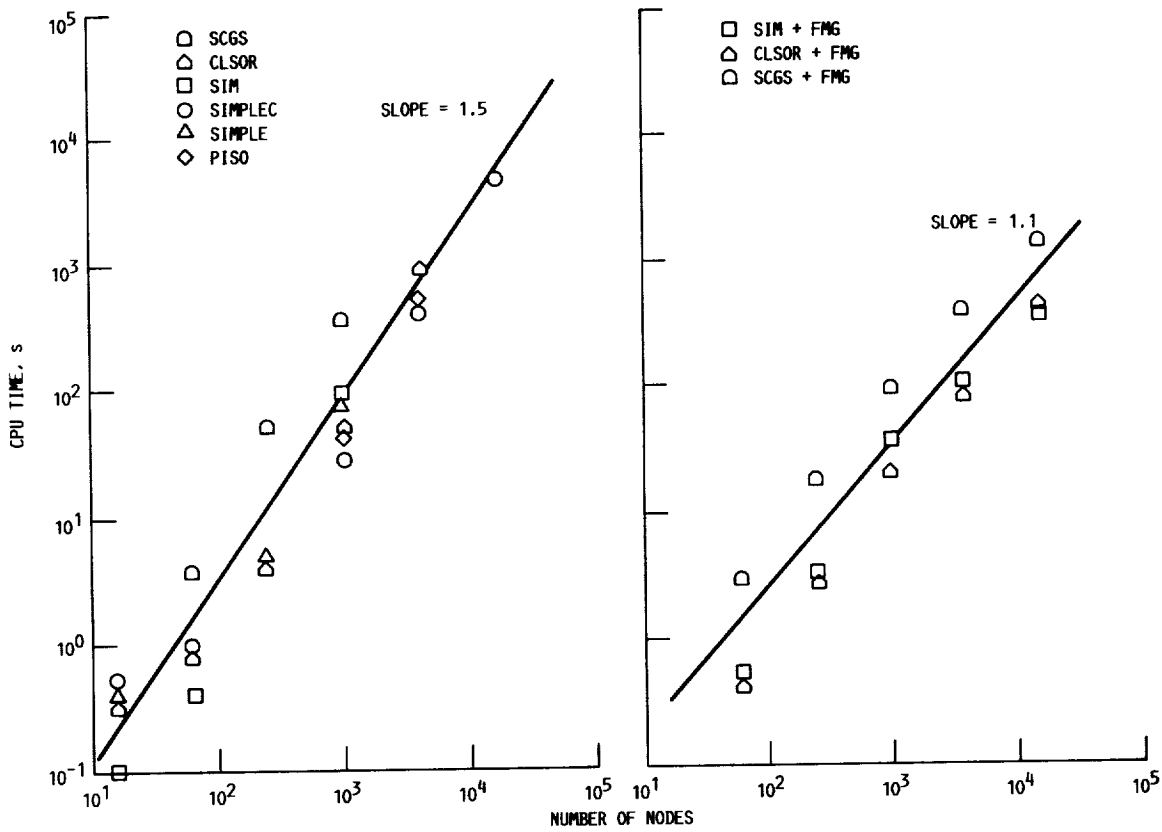


FIGURE 8. - COMPARISON OF CPU TIME FOR CONVERGENCE; BACKWARD FACING STEP FLOW,  $Re = 500$ .



# Report Documentation Page

1. Report No. <b>NASA TM-102359 ICOMP-89-24</b>		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle <b>Application of Multi-grid Methods for Solving the Navier-Stokes Equations</b>				5. Report Date <b>October 1989</b>	
				6. Performing Organization Code	
7. Author(s) <b>A.O. Demuren</b>				8. Performing Organization Report No. <b>E-4655</b>	
				10. Work Unit No. <b>505-62-21</b>	
9. Performing Organization Name and Address <b>National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191</b>				11. Contract or Grant No.	
				13. Type of Report and Period Covered <b>Technical Memorandum</b>	
12. Sponsoring Agency Name and Address <b>National Aeronautics and Space Administration Washington, D.C. 20546-0001</b>				14. Sponsoring Agency Code	
15. Supplementary Notes <p>A.O. Demuren, Institute for Computational Mechanics in Propulsion, Lewis Research Center (work funded under Space Act Agreement C99066G), and Faculty of Engineering, University of Lagos, Lagos, Nigeria. Space Act Monitor: Louis A. Povinelli.</p> <p>This paper is reproduced from the Proceedings of the Institution of Mechanical Engineers, Part C, 1989, by permission of the Council of the Institution.</p>					
16. Abstract <p>This paper presents the application of a class of multi-grid methods to the solution of the Navier-Stokes equations for two-dimensional laminar flow problems. The methods consist of combining the full approximation scheme—full multi-grid technique (FAS-FMG) with point-, line-, or plane-relaxation routines for solving the Navier-Stokes equations in primitive variables. The performance of the multi-grid methods is compared to that of several single-grid methods. The results show that much faster convergence can be procured through the use of the multi-grid approach than through the various suggestions for improving single-grid methods. The importance of the choice of relaxation scheme for the multi-grid method is illustrated.</p>					
17. Key Words (Suggested by Author(s)) <b>Multi-grid methods Navier-Stokes equations Finite-volume method Relaxation methods</b>			18. Distribution Statement <b>Unclassified – Unlimited Subject Category 64</b>		
19. Security Classif. (of this report) <b>Unclassified</b>		20. Security Classif. (of this page) <b>Unclassified</b>		21. No of pages <b>22</b>	22. Price* <b>A03</b>

