AIAA Journal
Vol. 31, No. 10, October 1993

Multigrid Acceleration of a Fractional-Step Solver in Generalized Curvilinear Coordinate Systems

Moshe Rosenfeld*
Tel Aviv University, 69 978 Tel Aviv, Israel
and
Dochan Kwak†
NASA Ames Research Center, Moffett Field, California 94035

A fractional-step (FS) solver of the three-dimensional time-dependent incompressible Navier-Stokes equations in generalized curvilinear coordinate systems, previously developed by the present authors, has been significantly enhanced by accelerating the Poisson solver with multigrid (MG) procedures. The most CPU time-consuming part of fractional-step methods is the solution of a discrete Poisson-like equation with Neumann-type boundary conditions formulated to satisfy the continuity equation. Usually, more than 80% of the total computational time of FS methods is consumed by the iterative solution of the Poisson equation. In the present study, multigrid techniques have been employed for accelerating the convergence rate of the Poisson solver in nonorthogonal coordinate systems. Various MG strategies have been tested in numerous numerical experiments. The total computational time required for solving the Poisson equation was reduced by an order of magnitude, whereas the overall computational time of the flow solver was reduced by a factor of 3–4. The MG Poisson solver consumes less than 25% of the total CPU time. The computational work has been found to be of order $\mathcal{O}(N)$, where N is the total number of mesh points, whereas the CPU time on a vector computer (CRAY Y-MP) is of $\mathcal{O}(N^{0.75})$. Consequently, the present method is a viable alternative for solving complex flowfields with a very large number of mesh points.

I. Introduction

MODERN computational fluid dynamics (CFD) simulations require very large numbers of mesh points to resolve turbulent flows over complicated configurations. The calculation of time-dependent incompressible flows poses an even greater computational challenge because a set of elliptic equations must be solved at each time step. Therefore, the enhancement of the efficiency of existing solution techniques of incompressible flows is an important issue.

Most solution algorithms of the time-accurate incompressible Navier-Stokes equations with primitive variables are related to the artificial compressibility (AC)[1,2] or the fractional-step (FS) methods.[3,4] In the AC methods, time accuracy is achieved by introducing subiterations for solving the discrete equations at each time level.[5] As in most iterative methods, the convergence rate of the subiterations deteriorates with the number of mesh points. This is an undesirable property, especially when large numbers of mesh points are required. The situation is similar with the FS solution methods. Here, a Poisson equation is formulated at each time step and is solved iteratively. Although in this case the iterations are usually performed on a single equation rather than on a set of equations as in the AC method, the convergence rate still deteriorates with the number of mesh points.

Direct solution of the discrete equations might resolve the problems associated with the convergence rate of iterative schemes, but the demand for very high computational resources (especially in memory) excludes it from being a feasible alternative for solving three-dimensional problems.

The operation count of the best iterative solver of a set of N algebraic equations cannot be less than the order of $\mathcal{O}(N)$. The multigrid (MG) method[6] is the only solution procedure with a linear operation count applicable to generalized curvilinear coordinate systems. The suitability of MG methods for accelerating time-accurate artificial compressibility methods is questionable because of the hyperbolic nature of the equations governing the pseudotime iterations. In contrast, MG techniques are well developed for elliptic equations, similar to the Poisson equation obtained in FS methods. The solution of this equation is the most time-consuming stage of FS methods; therefore, any enhancement of its convergence rate has a major effect on the overall computational time.

Ghia et al.[7] solved the two-dimensional Poisson equation with all Neumann-type boundary conditions (as is usually the case in FS methods) over an orthogonal coordinate system using an MG method. The unknowns are defined at the node points, and special restriction operators have been devised to satisfy the compatibility condition on all the mesh levels. Rieger et al.[8] considered the two-dimensional nonorthogonal Poisson equation with Neumann boundary conditions. The equation was formulated in a conservative form with the variables defined at the center of each cell. This arrangement of the variables proved to be very convenient for satisfying the compatibility condition on all of the MG levels. Good convergence properties have been obtained with an expensive relaxation procedure based on the strongly implicit method. Eliasson[9] used an FS method similar to that used by Rosenfeld et al.[4] to solve two-dimensional cases. The MG procedure implemented by Eliasson was based on the method of Rieger et al.,[8] but used the less costly four-color Zebra scheme of Rosenfeld et al.[4] Yet the efficiency of this MG method was not satisfactory. In typical cases more than 80% of the total computational time was spent on the Poisson solver. Similar performance was obtained in Refs. 10 and 11 using the four-color Zebra scheme on a single mesh.

In the present study, the efficiency of the INS3D-FS code, previously developed by the present authors,[4,10,11] has been significantly enhanced by accelerating the Poisson solver with a multigrid method. To the best knowledge of the authors, this is the first implementation of the MG method for solving the three-dimensional discrete Poisson-like equation obtained in

Presented as Paper 92-0185 at the AIAA 30th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1992; received Jan. 13, 1992; revision received Feb. 25, 1993; accepted for publication March 10, 1993.

*Lecturer, Department of Fluid Dynamics, Faculty of Engineering.

†Research Scientist, Fluid Dynamics Division.

the derivation of FS solution methods for the incompressible Navier-Stokes equations in generalized nonorthogonal coordinate systems using a staggered arrangement of the variables.

II. Numerical Model

The integral forms of the Navier-Stokes and continuity equations are discretized by the finite volume procedure. The formulation, discretization procedure, and numerical solution method are described in Refs. 4, 10, and 11 and therefore will be discussed here only briefly. The details of the derivation of the discrete Poisson-like equation were not emphasized in previous publications, and this topic will be elaborated in the present section.

A. Discretization

A general nonorthogonal coordinate system (ξ, η, ζ) is defined (discretely) by

$$r = r(\xi, \eta, \zeta) \tag{1}$$

where $r = (x, y, z)^T$ is the Cartesian coordinate system. The computational domain (ξ, η, ζ) is divided into uniform primary cells. The center of each primary cell is marked by the indices i, j, k. The integral equations are approximate in the physical space over small computational cells of general shape (see Fig. 1). Each computational cell is defined by its volume V, and the face vectors $\boldsymbol{S}^l$, where $l = \xi, \eta$, or ζ. The vector $\boldsymbol{S}^l$ has the magnitude of the area of the face and a direction normal to it.

The discretization procedure leads to the definition of a new set of dependent variables, the volume-fluxes U^ξ, U^η, and U^ζ through the ξ, η, or ζ faces, respectively, instead of the velocity vector $\boldsymbol{u}$

$$U^\xi = \boldsymbol{S}^\xi \cdot \boldsymbol{u}, \qquad U^\eta = \boldsymbol{S}^\eta \cdot \boldsymbol{u}, \qquad U^\zeta = \boldsymbol{S}^\zeta \cdot \boldsymbol{u} \tag{2}$$

The volume fluxes are equivalent to the contravariant velocity components (multiplied by the proper volume of the cell) over a staggered grid. This selection of the unknowns has certain advantages in the case of curvilinear nonorthogonal coordinate systems.[4,10] The mass conservation equation reduces to the simple form

$$U^\xi_{i+½} - U^\xi_{i-½} + U^\eta_{j+½} - U^\eta_{j-½} + U^\zeta_{k+½} - U^\zeta_{k-½}$$
$$= D_{iv}(U^l) = 0 \tag{3}$$

The summation operator D_{iv} is a discrete divergence-like operator [the divergence operator itself is $(1/V)D_{iv}$].

The operator form of the discrete ξ momentum equation using a two-level time integration scheme is

$$V_{i+½}\Delta U^\xi = G + \theta_d \Delta t D_\xi(\Delta U^l) + \theta_r \Delta t R_\xi(\Delta P) \tag{4}$$

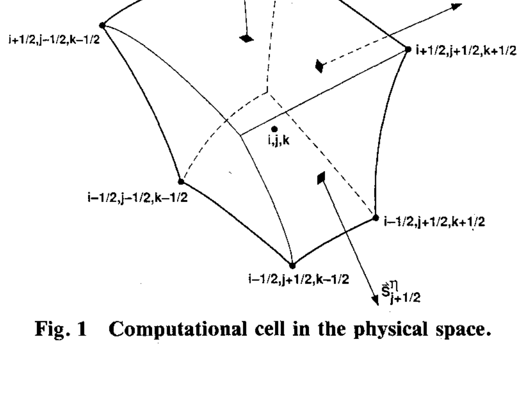


Fig. 1 Computational cell in the physical space.

where θ_d and θ_r are numerical parameters that determine the order of accuracy and the stability of the method, and Δt is the time step. The terms ΔU^ξ and ΔP are the delta form of the volume flux and the pressure P, respectively. The operator G includes all of the terms known from the previous two time levels. The operator D_ξ includes the implicit part of the convection and diffusion terms, whereas the operator R_ξ includes the pressure-gradient-like term given by

$$R_\xi(\Delta P) = -S^\xi_{i+½} \cdot (S^\xi_{i+1}\Delta P_{i+1} - S^\xi_i \Delta P_i$$
$$+ S^\eta_{i+½,j+½}\Delta P_{i+½,j+½} - S^\eta_{i+½,j-½}\Delta P_{i+½,j-½}$$
$$+ S^\zeta_{i+½,k+½}\Delta P_{i+½,k+½} - S^\zeta_{i+½,k-½}\Delta P_{i+½,k-½}) \tag{5}$$

The η and ζ momentum equations can be obtained by cyclic permutations.

B. Fractional Step Method

In the present implementation of the fractional step method, first the momentum equations are solved for an approximate volume flux in delta form $\Delta\tilde{U}^l$ by dropping $R_l(\Delta P)$ from Eq. (4), so that the pressure gradient is taken from the previous time level

$$(V_{m+½}I - \theta_d \Delta t D_l)\Delta\tilde{U}^l = G \tag{6a}$$

where $m = i, j$, or k for $l = \xi, \eta$, or ζ, respectively, and I is the identity operator. The momentum equations are solved using an implicit approximate factorization method.

The resulting flowfield does not generally satisfy the mass conservation equation, so that in the second step ΔU^l is calculated from

$$V_{m+½}(\Delta U^l - \Delta\tilde{U}^l) = \theta_r \Delta t R_l(\Delta P) \tag{6b}$$

to satisfy the continuity equation (cast in terms of ΔU^l) at the new time level $n+1$

$$D_{iv}(U^l)^{n+1} = 0 \tag{6c}$$

where D_{iv} is the summation operator defined in Eq. (3).

Equations (6b) and (6c) are combined into a single discrete Poisson-like equation by applying the discrete divergence-like operator D_{iv} on the discrete gradient-like operator R_l

$$-D_{iv}(\tilde{U}^l) = \theta_r \Delta t D_{iv}\left[\frac{R_l(\Delta P)}{V_{m+½}}\right] \tag{6d}$$

The operator $-(1/V)D_{iv}(R/V)$ is the discrete Laplacian-like operator. The present article considers only that second stage of the FS method.

C. Poisson-Like Equation

The discrete Poisson equation, Eq. (6d), can be rewritten as

$$L(\Delta P) = f \tag{7a}$$

where the source term f is related to the divergence of the velocity field

$$f = -\frac{D_{iv}(\tilde{U}^l)}{\theta_r \Delta t} \tag{7b}$$

If Eqs. (6b) and (6c) are to be satisfied exactly, the Laplacian-like operator L should be evaluated discretely from

$$L(\Delta P) = D_{iv}\left[\frac{R_l(\Delta P)}{V_{m+½}}\right] \tag{7c}$$

rather than from the differential operator. The approximation of L at each point involves 18 adjacent points.

The normal-derivative (Neumann) boundary condition for each boundary l is derived from Eq. (6b)

$$\theta_r \Delta t R_l(\Delta P) = V_{m+1/2}(\Delta U^l - \Delta \tilde{U}^l) \tag{8}$$

If Dirichlet-type boundary conditions are specified for the velocities, condition (8) is homogeneous.

The discrete set of the algebraic Eq. (7a) is singular for Neumann-type boundary conditions specified over all of the boundaries. Green's theorem applied to the Poisson equation $\nabla^2\phi = F$,

$$\oint_V \nabla^2\phi \, \mathrm{d}V = \oint_V F \, \mathrm{d}V = \oint_S \frac{\partial\phi}{\partial n} \mathrm{d}S \tag{9}$$

should be satisfied in the continuous case to ensure the existence and uniqueness (up to a constant) of the solution (here n is the normal direction to the boundary). In the present discrete formulation, this condition takes the form

$$\sum_{\text{all cells}} D_{iv}\left[\frac{R_l(\Delta P)}{V}\right] = \sum_{\text{all cells}} f = \sum_{\text{boundary faces}} \left[\frac{R_l(\Delta P)}{V}\right] \tag{10}$$

It should be noted that $R_l(\Delta P)/V = S^l \cdot \nabla(\Delta P) = S^l \partial(\Delta P)/\partial l$ and that $F = f/V$. But because in the present discretization procedure

$$\sum_{\text{all cells}} D_{iv}\left[\frac{R_l(\Delta P)}{V}\right] = \sum_{\text{boundary faces}} \left[\frac{R_l(\Delta P)}{V}\right] \tag{11a}$$

and [see Eq. (7b)]

$$\theta_r \Delta t \sum_{\text{all cells}} f = -\sum_{\text{all cells}} \tilde{U}^l = -\sum_{\text{boundary faces}} \tilde{U}^l \tag{11b}$$

are satisfied identically in the discrete case, Eq. (9) takes the form

$$\sum_{\text{boundary faces}} \left[\tilde{U}^l + \frac{\theta_r \Delta t R_l(\Delta P)}{V}\right] = 0 \tag{12}$$

In the case of homogeneous Neumann boundary conditions $[R_l(\Delta P) = 0]$, this "compatibility" condition requires the conservation of the discrete global mass rate flow $\Sigma\tilde{U}^l = 0$. The satisfaction of Eq. (12) is imposed by modifying the outflow boundary condition on U^l. Thus the existence of a solution is guaranteed without adding correction terms to the source f, as is usually the practice in other Poisson-equation-based solvers. The modification of the source term is equivalent to introducing a mass source term and results in loss of accuracy of the computed flowfield as well as adverse effects on the convergence properties of the Poisson solver.

It should be noted that the present formulation of the discrete Laplacian-like operator $D_{iv}(R_l/V)$ is essential for the satisfaction of the identities in Eqs. (11). A direct evaluation of the differential Laplacian operator does not usually satisfy these relations.

Before the solution method of the Poisson-like equation is described, several points should be stressed again.

1) The equation is derived *discretely*. Although it resembles the difference equations that would have been obtained from the discretization of a Poisson equation, the differential form of Eq. (7a) has no meaning in the FS method.

2) The accuracy of the discrete mass conservation equation is related to the accuracy of the solution of the discrete Poisson-like equation.

3) The satisfaction of the discrete compatibility condition [Eq. (12)] is a necessary and sufficient condition for the existence and uniqueness of the solution (up to an arbitrary constant). It also has favorable effects on the convergence properties of any iterative solution method.

4) Each discrete equation involves 19 points.

III. Multigrid Method

The reader is assumed to be familiar with multigrid methods and notation, see for example Refs. 6 and 12. The present section will describe only the implementation of the method for solving the set of the algebraic equations [Eq. (7)]. The correction scheme has been used in the present linear problem.

A. Relaxation

For a general nonorthogonal coordinate system, the 19-point-based discrete equations pose difficult challenges in obtaining vectorizable schemes. The present relaxation method uses a Zebra method with four-color ordering to decouple the unknowns and permit an effective use of vector computers. The three-dimensional Zebra relaxation scheme solves implicitly all of the equations along one coordinate line, say along ξ, as in the successive line over-relaxation (SLOR) method. However, the order in which the lines are processed is not the usual lexicographic order (by rows or columns), but a "colored" order. Existing two-dimensional applications of colored schemes use a two-color ordering (red-black schemes). This ordering is inappropriate for vectorizing nonorthogonal cases. In the present method, the points in the (η, ζ) plane are classified into four groups and a different color label is given to each group (see Fig. 2). First, all of the black lines are swept in a lexicographic order, then the red, blue, and green lines. The implicit solution of a line (along the ξ direction) is decoupled from the same color lines. For example, when a black line is being solved for, all of the neighboring lines are of different colors—red, blue, or green. This arrangement allows an efficient vectorization of the method, even with the 19-point computational stencil. The implicit solution along the ξ coordinate line may be exploited to enhance smoothing properties of problems with heavily clustered mesh points along that direction.

The Zebra scheme has good smoothing properties and in many cases is nearly optimal for using a multigrid acceleration procedure.[12]

In our previous works, this method has been used to solve the Poisson-like equation using a single grid. To enhance its convergence rate, an over-relaxation parameter has been experimentally selected for each case. For comparison purposes, this single-grid method is also used in the present study.

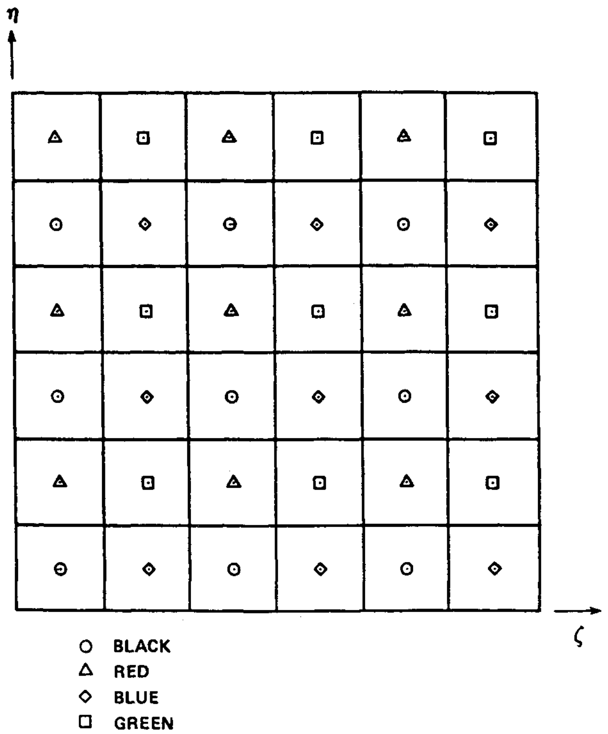


Fig. 2 Arrangement of the colors in the (ξ, η) plane.

B. Coarsening Method

The term coarsening method refers to the method of obtaining the coarse grids from a given fine grid. In the standard MG method, the coarse grids are obtained by halving the number of mesh points in each direction. This method is termed full coarsening. In many circumstances, especially in three-dimensional cases with generalized curvilinear coordinate systems, semicoarsening may be preferable. In the semicoarsening method, the mesh points are halved only along some of the coordinate directions to enhance the efficiency of the MG procedure.[12] Both full- and semicoarsening methods (along one or two coordinate directions) are implemented.

A key feature of the present MG implementation is the cellwise coarsening procedure, rather than the more common pointwise coarsening. In the full-coarsening approach, the coarse grid is constructed from eight adjacent fine cells (see Fig. 3). The volume of the coarse cell is calculated as the sum of the volumes of the individual fine grid cells:

$$V^{m-1}_{I,J,K} = V^m_{i,j,k} + V^m_{i+1,j,k} + V^m_{i,j+1,k} + V^m_{i,j,k+1} + V^m_{i+1,j+1,k} + V^m_{i,j+1,k+1} + V^m_{i+1,j,k+1} + V^m_{i+1,j+1,k+1} \tag{13}$$

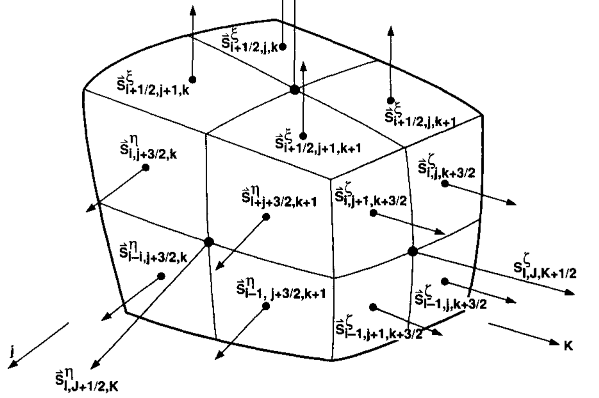
The indices I, J, and K refer to the coarse grid, whereas i, j, and k refer to the fine grid and m is the mesh-level index. The two set of indices are related by $i = 2I - 1$, $j = 2J - 1$, and $k = 2K - 1$. Similarly, the faces of the coarse cell are constructed from the external faces of the combining fine cells. Thus, the S^ξ face of the coarse grid is given by

$$S^\xi_{I+½,J,K} = S^\xi_{i+½,j,k} + S^\xi_{i+½,j+1,k} + S^\xi_{i+½,j,k+1} + S^\xi_{i+½,j+1,k+1} \tag{14}$$

Here, the mesh-level index is omitted for clarity. The most important property of this procedure of cellwise coarsening is the automatic satisfaction of the geometry conservation laws[4] on all the multigrid levels. Equivalent procedures are used in the semicoarsening methods.

C. Restriction

The discrete equations to be solved are singular, because of the all-Neumann-type boundary conditions. Any restriction operator should be devised so that the compatibility condition [Eq. (12)] is satisfied on all of the levels. The derivative boundary conditions are treated formally as source terms by including them in the right-hand side of Eq. (7a). Thus, the compatibility condition requires that the sum of the source terms (the residuals plus the boundary conditions) vanishes on all of the levels.



$\vec{S}^\xi_{I+1/2,J,K}$
$\vec{S}^\xi_{i+1/2,j,k}$
$\vec{S}^\xi_{i+1/2,j+1,k}$
$\vec{S}^\xi_{i+1/2,j,k+1}$
$\vec{S}^\xi_{i+1/2,j+1,k+1}$
$\vec{S}^\eta_{i,j+3/2,k}$
$\vec{S}^\eta_{i+j+3/2,k+1}$
$\vec{S}^\zeta_{i,j+1,k+3/2}$
$\vec{S}^\zeta_{i,j,k+3/2}$
$\vec{S}^\eta_{i-i,j+3/2,k}$
$\vec{S}^\eta_{i-1,j+3/2,k+1}$
$S^\zeta_{I,J,K+1/2}$
$\vec{S}^\zeta_{i-1,j,k+3/2}$
$\vec{S}^\zeta_{i-1,j+1,k+3/2}$
j
K
$\vec{S}^\eta_{I,J+1/2,K}$

Fig. 3 Coarse cell in the full coarsening method.

A first-order restriction operator that satisfies this condition can be formulated straightforwardly in the present finite-volume discretization procedure with cellwise coarsening. In the case of full coarsening, the restriction operator I^{m-1}_m is defined as the sum of the residuals of the eight fine cells that combine the coarse cell (I, J, K)

$$R^{m-1}_{I,J,K} = I^{m-1}_m R^m_{i,j,k} = R^m_{i,j,k} + R^m_{i+1,j,k} + R^m_{i,j+1,k} + R^m_{i,j,k+1} + R^m_{i+1,j+1,k} + R^m_{i,j+1,k+1} + R^m_{i+1,j,k+1} + R^m_{i+1,j+1,k+1} \tag{15}$$

where R is the residual. It is easy to verify that this choice of the restriction operator satisfies the compatibility condition on the coarser grids as well.

$$\sum_{I,J,K} R^{m-1}_{I,J,K} = \sum_{i,j,k} I^{m-1}_m R^m_{i,j,k} = \sum_{i,j,k} R^m_{i,j,k} = 0 \tag{16}$$

A second-order restriction operator that satisfies the compatibility equation has been derived as well. Yet numerical experiments (see Results) showed little sensitivity of the MG method in the accuracy of the restriction operator. Hence the first-order restriction operator is preferred because it saves computational work.

D. Interpolation

The coarse grid unknowns are not a subset of the fine grid unknowns because they are defined at the center of the computational cells. In the full coarsening of three-dimensional cases, a trilinear interpolation in the computational space is used for transferring the coarse grid correction into the proper location on the next fine grid. Similarly, in the semicoarsening cases, bilinear or linear interpolations are used, depending on the semicoarsening method. Interpolation in the physical space had no effect on the convergence rate, but it required larger computational effort and therefore was abandoned.

The combined order of the restriction and interpolation operators is three, which is adequate for the second-order Poisson equation.[6]

E. Cycling Strategies

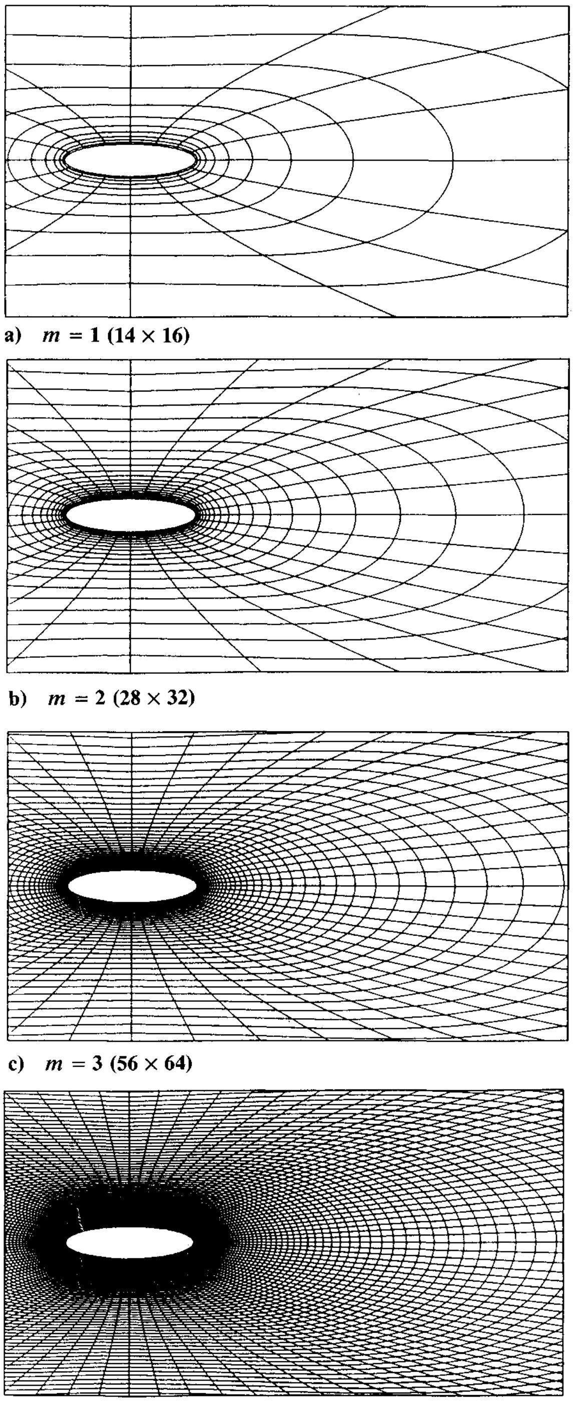
The present implementation uses both accommodative and fixed cycling procedures for solving the equations by MG. In the first category, the C cycle has been used,[6] whereas in the second category a generalized MG cycle has been implemented.[12] Depending on the value of the parameter γ_m, which is a function of the mesh level m (see Ref. 12 for notation), the V($\gamma_m = 1$) or the W($\gamma_m = 2$) cycle can be obtained as a special case.

Both the standard- and the full-multigrid (FMG) methods can be used with the generalized MG cycle. In the present context of time-accurate solution of the Navier-Stokes equation by an FS method, the solution of the discrete Poisson equations should converge to a good accuracy for yielding divergence-free velocity fields. The FMG cycle may not reduce the error sufficiently, because the MG theory predicts the resulting error to be of the order of the truncation errors. However, in the present method the truncation errors are not relevant because a continuous Poisson equation is never derived nor solved. If necessary, after a single FMG cycle is completed, additional MG cycles are performed until the convergence criterion is met.

The FMG method starts the MG solution at the coarsest grid. In the present study, the discrete Poisson-like equation is formulated on the finest grid only. To define the equivalent of this equation on the coarsest grid, the source term f is restricted into the coarsest level, using the restriction operator [Eq. (15)] to move down the grids.

IV. Results

This section describes some of the numerical experiments performed in the implementation of the MG method. It should be stressed that neither the accuracy nor other numerical properties of the FS method itself have been changed. The contribution of the present work is in accelerating the convergence rate of the Poisson solver, and only this aspect will be considered. Three of the test cases will be described here. Many other cases have been solved with similar efficiency using the MG procedures. The first two test cases employ two-dimensional nonorthogonal coordinate systems to demonstrate the performance of the MG methods. One case considers the domain over an elliptic airfoil, and the other case solves the equations between two eccentric cylinders with different values of eccentricity. The three-dimensional test case considers a square duct with a 90-deg bend. All of these cases represent problems that have been solved to investigate time-dependent incompressible viscous flows.


a) $m = 1$ (14 × 16)

b) $m = 2$ (28 × 32)

c) $m = 3$ (56 × 64)

d) $m = 4$ (112 × 128)

Fig. 4 Four levels in the vicinity of the elliptic airfoil.

A. Two-Dimensional Elliptic Airfoil

Table 1 lists the results of a parametric study performed for the case of a flow over an elliptic airfoil with a thickness ratio of 1:4. A generalized nonorthogonal curvilinear coordinate system is employed with 112 × 128 mesh points (in the third direction three mesh points are specified). This two-dimensional case is coarsened along the ξ and η directions and a total of four mesh levels have been used, with the coarsest mesh consisting of 14 × 16 mesh points (see Fig. 4). The convergence criterion (based on the maximal residual of the Poisson equation) is set to 10^{-9}. The parameter ν refers to the number of pre- and postrestriction iterations; Res. is the accuracy order of the restriction (first- or second-order accurate); cycle is the number of cycles performed; and WU is the total number of work units. A WU is defined as the computational work required to perform one complete relaxation iteration on the finest mesh. It does not take into account the computational work required for performing the grid transfers and the work for solving the equations on the coarsest grid. The overhead of these calculations usually increases the WU count by 15–30%. The average convergence rate per work unit μ is defined by

$$\mu = \left(\frac{\epsilon}{\epsilon_{\text{ini}}}\right)^{1/(\text{WU} - 1)} \tag{17}$$

where ϵ and ϵ_{ini} are the error and the initial error (based on the maximal residual), respectively. In the table, CPU is the CPU time in seconds on the CRAY Y-MP/832.

Table 1 reveals that the effect of the order of accuracy of the restriction operator is negligible. The optimal number of relaxation sweeps seems to be $\nu = 2$. A significant reduction in CPU time and number of WU can be obtained by selecting the cycling sequence $\gamma_m = (2, 1, 2, 1)$. It is obvious that the CPU time is not proportional to the number of WU. This relation is a result of using a vector computer. The iterations performed on the coarser grids are not as efficient as the iterations on the finest grid. Therefore, although the WU count of case 5 is almost twice that of case 7, the CPU time increased only 1.28 times.

Table 2 compares the performance of several MG methods and the Zebra single-grid method for two different convergence criteria ($\epsilon = 10^{-9}$ and $\epsilon = 10^{-11}$) where Mflops is million floating point operations per second on the CRAY Y-MP. In the single-grid method the four-color Zebra method was used

Table 1 Effect of MG parameters (elliptical airfoil)

Case	ν	γ_m	Res.	Cycle	WU	μ	CPU,s
1	2	1,1,1,1	2	5	27.3	0.814	1.22
2	2	1,1,1,1	1	5	27.3	0.808	1.19
3	1	1,1,1,1	1	12	32.5	0.850	1.68
4	3	1,1,1,1	1	4	32.5	0.834	1.26
5	5	1,1,1,1	1	3	40.4	0.870	1.45
6	2	1,2,1,2	1	4	23.0	0.773	1.18
7	2	2,1,2,1	1	3	20.5	0.752	1.06
8	2	2,2,2,2	1	3	22.0	0.746	1.26

Table 2 Efficiency of cycling methods

Cycle	WU	μ	CPU, s	Mflops
	$\epsilon = 10^{-9}$			
Full MG	8.0	(0.791)	0.50	47
Standard MG	7.5	0.743	0.42	51
Single grid	50	0.919	1.18	
	$\epsilon = 10^{-11}$			
Full MG	34.0	(0.863)	1.57	62
Standard MG	33.5	0.819	1.41	65
C cycle	35.8	0.830	1.73	68
Single grid	350	0.979	7.14	97

to solve the equations iteratively. The convergence of the method was optimized by using an over-relaxation factor determined by numerical experiments. This method will be referred to as the SOR method because the convergence properties are similar to the SLOR method. The standard MG procedure is found to be the fastest alternative in both cases, whereas the iterative solver without MG (the SOR method) showed the poorest performance, especially in the case of the finer convergence criterion. The ratio between the CPU time consumed by the single-grid method and the standard MG method is 2.8 for $\epsilon = 10^{-9}$ and 5.1 for $\epsilon = 10^{-11}$. The corresponding ratios of WU, 6.7 and 10.4, respectively, show an even better performance of the MG method. The use of the coarse grids in the MG method degrades the CPU time gain on vector computers. Clearly, Table 2 demonstrates that the single-grid method runs at a rate of 97 Mflops, whereas the standard MG solver runs at a rate of only 65 Mflops. It indicates that the optimization of MG algorithms on vector machines should favor relaxations on finer grids or limit the minimal size of the coarsest grid.

Figure 5 shows the convergence history of the Poisson solver for two single-grid iterative methods (the Zebra scheme without and with over-relaxation) and three MG cycling procedures (C cycle, standard, and full MG). The C cycle is slightly less efficient than the other two MG solvers. The single-grid solvers are much less efficient. Only the Zebra method with optimal over-relaxation parameter (the SOR method) converges to a reasonable degree, although the convergence rate is poor.

B. Two-Dimensional Eccentric Cylinders

To systematically study the properties of the proposed MG procedures, the discrete Poisson equation [Eq. (7)] is solved in a domain between two eccentric circular cylinders. The diameters of the inner and outer cylinders are 1 and 21, respectively. An algebraic nonorthogonal coordinate system has been generated with mesh points clustered in the vicinity of the inner cylinder and at the "wake" region. Figure 6 shows plots of the resulting grids for four different values of the eccentricity, $e = 0$, 6, 12.5, and 19. Here, the eccentricity is defined as the distance between the centers of the two circular boundaries. For clarity, a coarse mesh of 32 × 32 points is shown. The increase in eccentricity yields less orthogonal meshes with larger ratios between the maximal and minimal Jacobians.

Figure 7 gives the convergence history of the SOR and the MG procedures for a case with eccentricity of 12.5, using a grid of 256 × 256 mesh points with a total of seven multigrid levels. In Fig. 7a the convergence history is plotted against the number of WU, and Fig. 7b shows the dependence on the CPU time. Each of the solution methods has been separately optimized to yield the best convergence rate. On a vector computer, the measures of WU and CPU time are not equivalent. WU is essentially proportional to the operation count, whereas the CPU time consumption on a vector computer depends not only on the number of algebraic operations, but also on other factors such as the length of the vectors. Therefore, in Fig. 7 and following figures, WU is a measure of the performance on scalar machines, whereas CPU is a measure of its performance on vector machines.

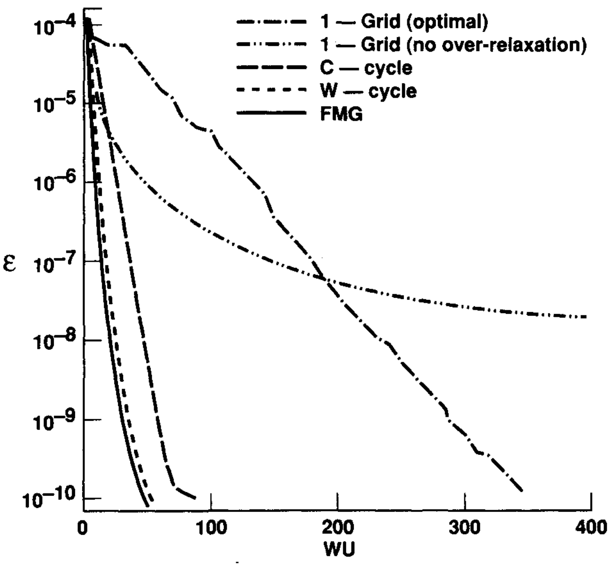
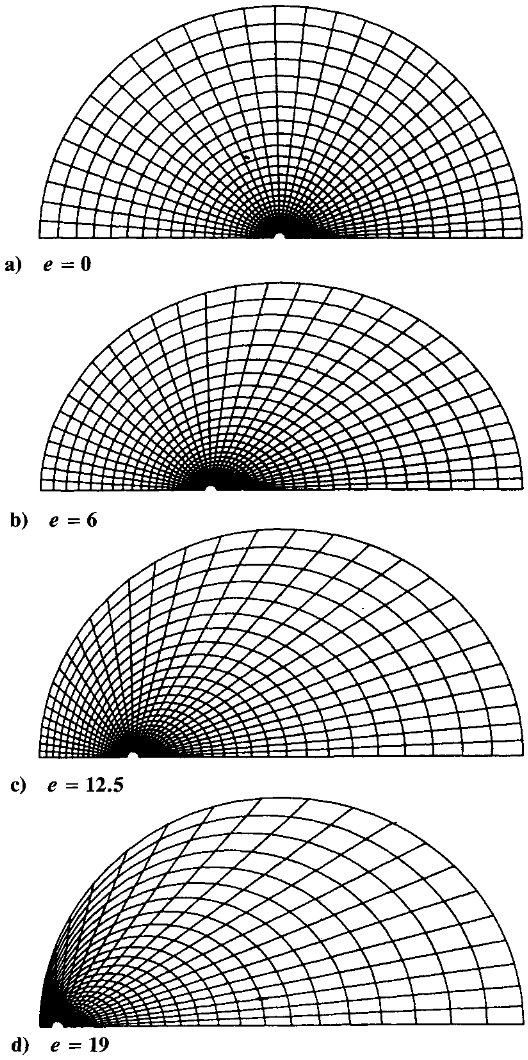


Fig. 5 Convergence history of MG cycling methods (elliptic airfoil).


a) $e = 0$

b) $e = 6$

c) $e = 12.5$

d) $e = 19$

Fig. 6 Meshes with different eccentricity values.

The difference between the two measures is clearly observable in Figs. 7a and 7b. The C cycle has a better convergence rate than the SOR method in terms of WU. The contrary is true if the convergence rate is based on the CPU time. Most of the calculations performed in the C cycle are done on coarser grids, with excessive time spent on relatively inefficient grid transfers. By any measure, the fixed MG cycles outperform the optimal single-grid method. The FMG method is 9.4 times faster in CPU time and requires 17.5 times fewer WU. Not surprisingly, the average convergence rates μ of the standard and full MG cycles are almost identical with the convergence rate found in the previous test case of an elliptic airfoil.

The case with eccentricity of 12.5 has also been used to study the dependence of the total work required to solve a problem (within a specified convergence error) on the number of mesh points. Figure 8a shows the total number of WU required to solve the equations on successively finer grids with 1) 32 × 32, 2) 64 × 64, 3) 128 × 128, 4) 256 × 256, and

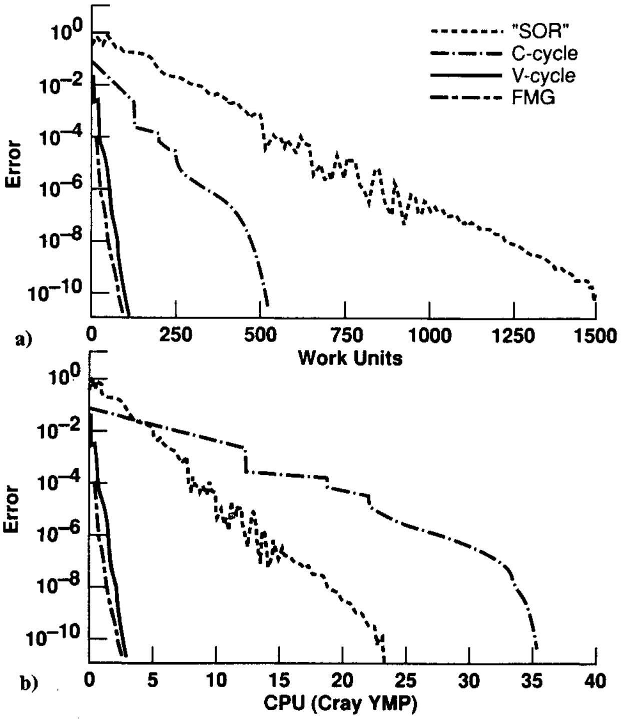


Fig. 7 Convergence history of MG cycling methods (e = 12.5).

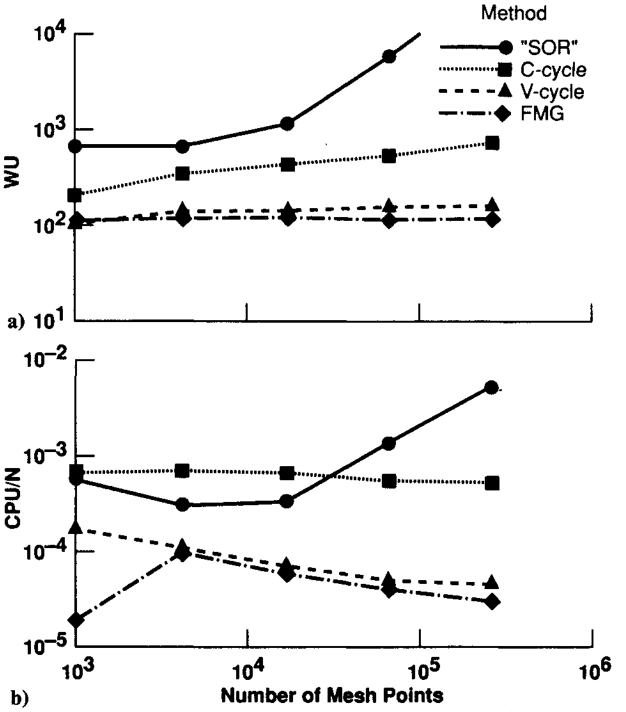


Fig. 8 Dependence of a) total WU and b) CPU/N on the mesh size.

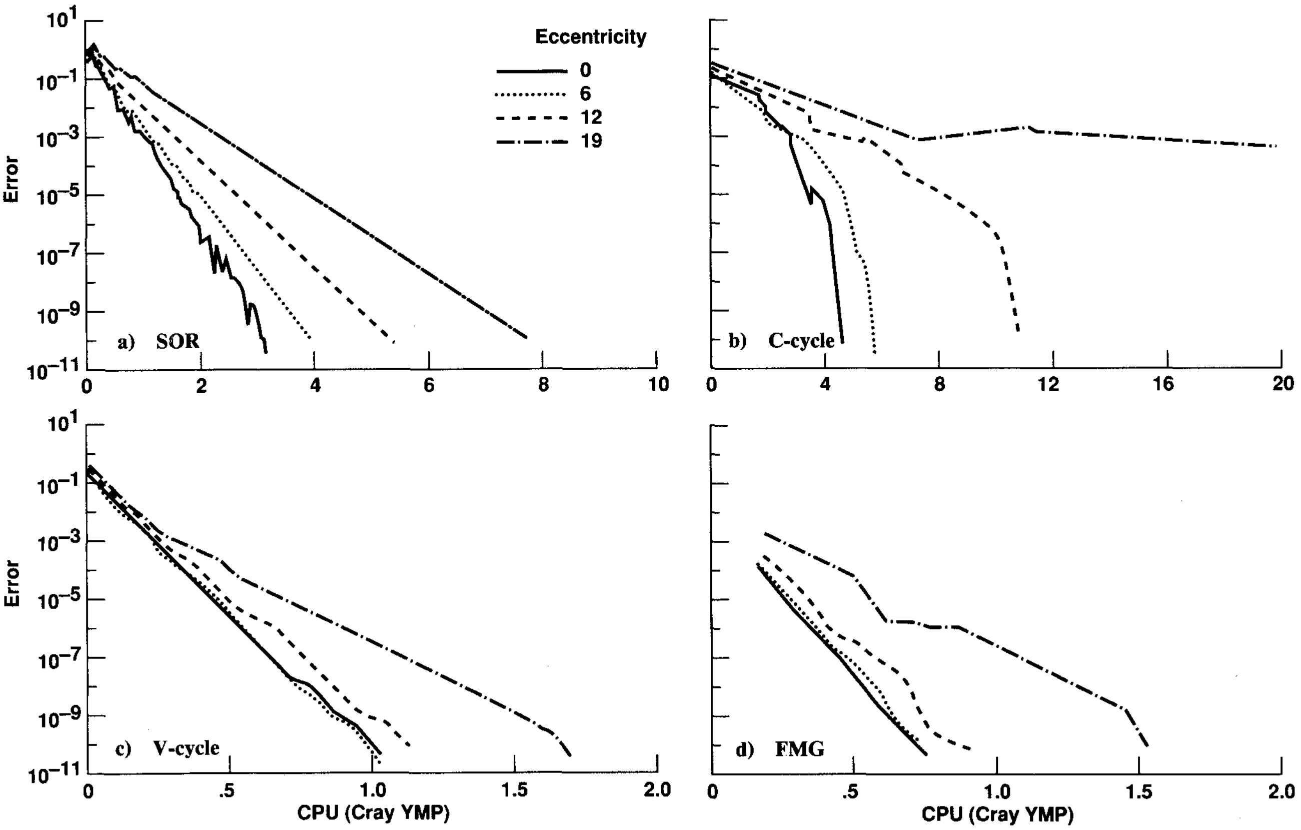


Fig. 9 Effect of geometry on the convergence properties.

5) 512 × 512 mesh points. In all of the cases the coarsest grid consists of 4 × 4 mesh points, and the number of grid levels varies, from four, grid 1, up to eight for the finest grid. The solution methods were optimized for mesh 3, and the same parameters were used for all of the other meshes to test the robustness and sensitivity of the methods to the value of the control parameters. It should be kept in mind that better performance of the SOR method and to some extent of the C-cycle MG procedure could have been obtained if the control parameters (especially the over-relaxation parameter) had been optimized for each grid separately.

The SOR method demonstrates the most severe degradation with the increase in the number of mesh points. The C cycle shows a weak dependence on the number of mesh points. The V cycle and especially the FMG method converge on all of the grids within a constant number of WU, proving the theoretical result that MG methods have a work count of order $\mathcal{O}(N)$, where N is the total number of mesh points. This is a property of significant implications for the efficiency of FS methods on very fine grids.

Figure 8b gives the CPU time divided by the number of mesh points (CPU/N) required to converge the solution on each of the grids. This measure is equivalent to the number of WU on a scalar machine. Such is not the case on vector machines where large grids (longer vectors) are favored over coarser meshes. The FMG and V-MG cycles show an enhanced performance on the finer grids. The CPU time required for solving the Poisson equation (in the range of the number of mesh points tested) is of $\mathcal{O}(N^{0.75})$ on the CRAY Y-MP instead the theoretical value of $\mathcal{O}(N)$ predicted by the MG theory. The gain is attributed solely to the properties of vector computers. The corresponding CPU times for the C-cycle MG method and the SOR single-grid method are $\mathcal{O}(N^{0.9})$ and $\mathcal{O}(N^2)$, respectively, making the SOR method impractical for very fine grids.

To study the sensitivity and robustness of the methods with geometry changes, the eccentricity of the grid is changed without changing the parameters that control the convergence rate of the Poisson solvers (the parameters have been optimized for the case of $e = 12.5$). Figure 9 shows the convergence history with the CPU time of the SOR method, and the MG cycles of types C, V, and FMG. The C cycle demonstrates the strongest dependence on geometry. In the case of the largest eccentricity ($e = 19$) the method stalls. The SOR method converges in all of the cases, yet the convergence rate deteriorates almost proportionally with the value of the eccentricity. The V cycle and the FMG method are the most robust methods. The convergence history is almost independent of the eccentricity for $e < 10$. Only at the highest eccentricity values do these methods deteriorate in the convergence properties, although convergence is achieved in all of the cases. Bearing in mind that, in realistic CFD calculations, meshes equivalent to grids with high eccentricity values are not used anyway, it can be concluded that the FMG and MG procedures using V cycles are insensitive to reasonable geometric changes. In particular, an optimized MG procedure can be used for a wide range of cases without changing the control parameters.

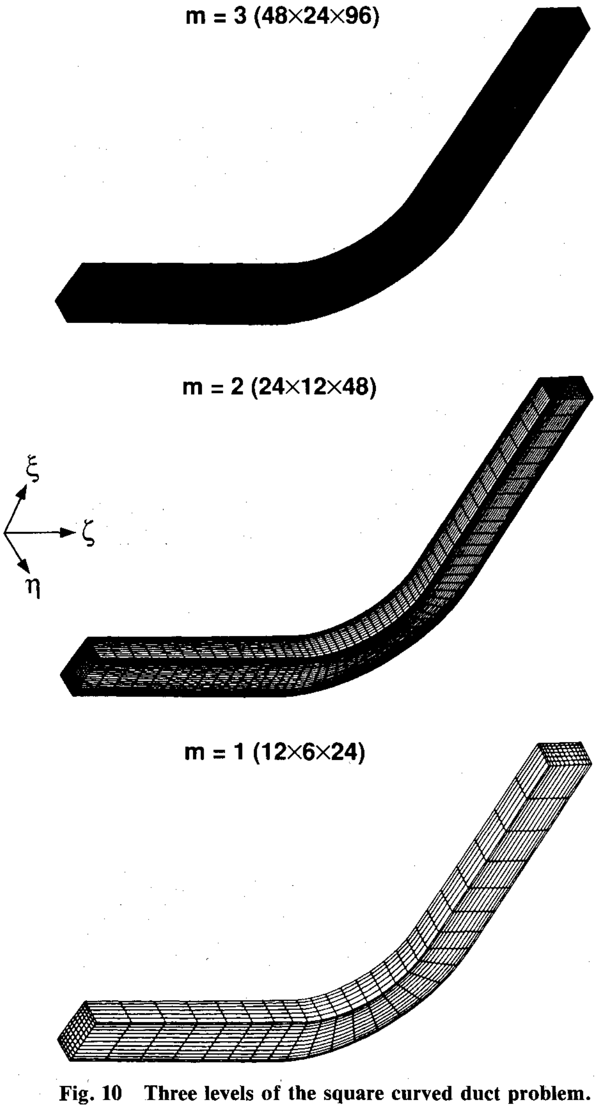


Fig. 10 Three levels of the square curved duct problem.

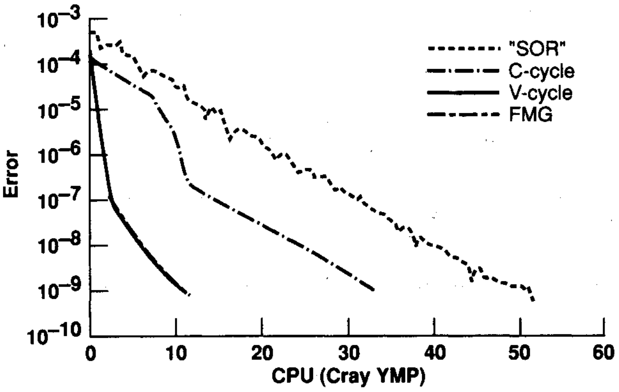


Fig. 11 Convergence history of MG cycling methods (duct case).

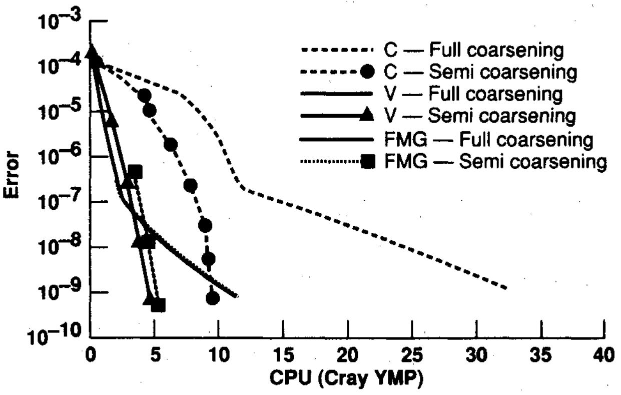


Fig. 12 Effect of the coarsening method on the convergence.

C. Three-Dimensional Square Duct with a Bend

As a three-dimensional test case, the discrete Poisson equation obtained in the solution of the flow in a three-dimensional curved square duct at a Reynolds number of 790 is solved using a mesh of 48 x 24 x 96 points. The symmetry of the problem was utilized to save mesh points in the η direction. Only three mesh levels have been used. Using full coarsening, the coarsest mesh consists of 6 × 12 × 24 mesh points. Figure 10 shows a view of the three mesh levels.

The convergence history of the optimized methods is given in Fig. 11. In this three-dimensional case the SOR method has the worst convergence rate, with regard to both WU and CPU time. It converges in 51.7 CRAY Y-MP CPU s, whereas the C-cycle MG procedure converges in 33.2 CPU s. The V cycle and the FMG have identical convergence properties; they converge in 11.3 s, which is 4.4 times faster than the optimal SOR method.

Yet all of the MG procedures show a nonuniform convergence history, with a significant decrease of the convergence rate after a few MG cycles. This may indicate a poor relaxation scheme or an inadequate coarsening method. Semicoarsening in the ξ and ζ or in the η and ζ directions results in an even less favorable convergence rate. However, semicoarsening in the ξ and η directions (leaving the direction along the duct axis uncoarsened) leads to a dramatic improvement, as shown in Fig. 12, which compares the performance of the full-coarsening and semicoarsening methods. A constant convergence rate is obtained over all of the MG cycles. Although the amount of work for each MG cycle is increased in the semicoarsening procedure, the total CPU time has decreased to 9.5 CPU s for the C cycle, and 4.6 and 5.3 CPU s for the V-type standard and full MG procedures, respectively.

In terms of WU, the gain of the MG procedure is even larger. The V cycle requires more than 28 times fewer WU than the SOR method, yet it consumes only 11.2 times less CPU time. The difference is mainly attributed to the considerably shorter vector length of the present three-dimensional case, because each dimension has a relatively small number of mesh points compared to the problems solved in the two-dimensional examples. But it also indicates that the possible gain in larger three-dimensional problems is large.

It should be recalled that only three mesh levels have been used, and therefore the MG methods have not reached their asymptotic convergence rate. For larger problems more mesh levels can be used, and the convergence rate is anticipated to improve.

V. Concluding Remarks

The implementation of the multigrid method for solving the discrete Poisson-like equation of a fractional-step method suggested by Rosenfeld et al.[4] proved that it is an efficient and robust solver. The MG solver was found to be insensitive to the geometry, even in cases of highly nonorthogonal clustered meshes, as well as to other parameters of MG methods. Optimizing an MG procedure for a particular problem was equally applicable to many other cases differing in grid properties and number of mesh points.

The total count of work units needed to solve the Poisson-like equation was found to be independent of the number of mesh points, proving that the MG solver is of order N operations, where N is the total number of mesh points. On the CRAY Y-MP vector machine, the total CPU time was found to be of order $N^{0.75}$ (for grids in the range of $N = 10^3\text{–}2.5 \times 10^5$ points). Because the operation count of the approximate factorization solver of the momentum equations is also proportional to the number of mesh points, the computational effort of the present FS method is linear with the number of mesh points. This feature is significant for future applications of the FS method to problems with very large numbers of mesh points, where other iterative solvers might be too expensive.

In typical cases solved, the MG solver is an order of magnitude faster than the single-grid four-color Zebra method. This results in a decrease of 300–400% of the CPU time for the complete FS solution. Without MG acceleration (but with optimal over-relaxation), the Poisson solver may consume more than 80% of the total time. With MG acceleration, the Poisson solver consumes less than 25% of the total computational time. For typical two-dimensional problems, the computational time per mesh point per time step is 30 μ CPU s on the CRAY Y-MP; for three-dimensional problems it is 160 μs.

Acknowledgment

The first author was partially supported by MCAT Institute, San Jose, California.

References

[1] Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics,* Vol. 2, No. 12, 1967, pp. 12–26.

[2] Kwak, D., Chang, J. L., Shanks, S. P., and Chakravarthy, S., "An Incompressible Navier-Stokes Flow Solver in Three-Dimensional Curvilinear Coordinate Systems Using Primitive Variables," *AIAA Journal,* Vol. 24, No. 3, 1986, pp. 390–396.

[3] Chorin, A. J., "Numerical Solution of the Navier-Stokes Equations," *Mathematics of Computation,* Vol. 22, No. 104, 1968, pp. 745–762.

[4] Rosenfeld, M., Kwak, D., and Vinokur, M., "A Fractional-Step Solution Method for the Unsteady Incompressible Navier-Stokes Equations in Generalized Coordinate Systems," *Journal of Computational Physics,* Vol. 94, No. 1, 1991, pp. 102–137.

[5] Rogers, S. E., and Kwak, D., "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady State and Time-Dependent Problems," AIAA Paper 89-0463, Jan. 1989.

[6] Brandt A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation,* Vol. 31, April, 1977, pp. 333–390.

[7] Ghia, U., Ramamurti, R., and Ghia, K. N., "Solution of the Neumann Pressure Problem in General Orthogonal Coordinates Using the Multigrid Technique," *AIAA Journal,* Vol. 26, No. 5, 1988, pp. 538–547.

[8] Rieger, H., Projahn, U., and Beer, H., "Fast Iterative Solution of Poisson Equation with Neumann Boundary Conditions in Nonorthogonal Curvilinear Coordinate Systems by a Multiple Grid Method," *Numerical Heat Transfer,* Vol. 6, No. 1, 1983, pp. 1–15.

[9] Eliasson, P., "A Solution Method for the Time-Dependent Navier-Stokes Equations for Laminar, Incompressible Flow," Aeronautical Research Institute of Sweden, FFA Rept. 146, Stockholm, Nov. 1989.

[10] Rosenfeld, M., Kwak, D., and Vinokur, M., "Development of an Accurate Solution Method for the Unsteady and Incompressible Navier-Stokes Equations in Generalized Coordinate Systems," NASA TM 103912, Nov. 1992.

[11] Rosenfeld, M., and Kwak, D., "Time-Dependent Solutions of Viscous Incompressible Flows in Moving Coordinates," *International Journal of Numerical Methods in Fluids,* Vol. 13, No. 10, 1991, pp. 1311–1328.

[12] Brandt, A., "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," GMD-Studien Nr. 85, GMD, Birlinghoven, Germany, May 1984.