

Comparison of Implicit Multigrid Schemes for Three-Dimensional Incompressible Flows

Li Yuan

LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and System Sciences, Academia Sinica, Beijing 100080, China

E-mail: lyuan@lsec.cc.ac.cn

Received April 25, 2001; revised September 4, 2001

To develop a robust and efficient computational flow simulation tool for incompressible flow applications, a number of different implicit multigrid schemes for solving the three-dimensional incompressible Navier–Stokes equations are compared in the current study. These schemes consist of a common full approximation storage (FAS) multigrid algorithm implemented in conjunction with three different implicit schemes, which include a modified point Gauss relaxation, a standard Gauss–Seidel line relaxation, and the Beam–Warming alternating direction implicit (ADI) scheme. The flow solver used in the study is based on artificial compressibility and uses a third-order upwind difference for the convective terms and a second-order central difference for the viscous terms. The efficiency of each implicit multigrid scheme is assessed in terms of the computing time required for two laminar flow problems: the entry flow through a 90° bent square duct, and the steady-state and unsteady flows past a prolate spheroid at incidence with an axis ratio of 4 : 1. It is found that implementation of Neumann boundary conditions on the coarse grid in terms of the flow variable correction rather than the flow variable itself is essential for obtaining good convergence in the collocated finite difference discretization. The results of steady-state flow computations show that all the implicit multigrid schemes yield more than 50% computational time savings over their single grid counterparts, and the point or line relaxation multigrid scheme outperforms the ADI multigrid scheme by at least a factor of 2. However, in unsteady flow computations, the computational time saving of the multigrid scheme is less than that in steady-state cases. The current study concludes that the FAS multigrid algorithm implemented with the modified point Gauss relaxation scheme is preferable for simulating both steady-state and time-dependent incompressible flows. © 2002 Elsevier Science (USA)

Key Words: multigrid; implicit scheme; artificial compressibility; incompressible Navier–Stokes equations; prolate spheroid.

1. INTRODUCTION

The computational simulation of three-dimensional (3D) incompressible flows is an area of intense research activity, as many realistic problems in low-speed aerodynamics and hydrodynamics can be addressed by the incompressible Navier–Stokes equations (INSE). Numerical solutions of the 3D INSE, nevertheless, require a very large number of grid points to resolve the flow structure and to obtain grid-independent solutions; thus they are computationally intensive. A further difficulty arises in solving the INSE because of the lack of a time-derivative term in the continuity equation, which limits the straightforward use of time-marching numerical methods. Most numerical methods [1–4] for the INSE involve solving a Poisson equation for the pressure, which is the most CPU-time-consuming part of these methods. They tend to be computationally expensive in general 3D curvilinear coordinates because the discretized pressure Poisson equation is sometimes troublesome on general curvilinear grids [5], whereas an accurate solution of the discretized Poisson equation must be performed at each time step. There have been some improvements in which multigrid (MG) techniques were implemented either to solve the pressure Poisson equation alone [6] or to solve all the coupled equations in a block-implicit manner to speed up convergence [7]. To achieve efficiency and robustness in simulating engineering flow problems, it is also desirable to use other cost-effective numerical methods with reasonable accuracy. The artificial compressibility method (ACM) [8], which avoids solution of the pressure Poisson equation, seems to be such a method. The main advantage of using the ACM is that it changes the whole set of governing equations into a hyperbolic–parabolic-type time-dependent system of equations by introducing a pseudo-time derivative of pressure into the continuity equation, hence its discretization and solution technique can be borrowed from some efficient methods developed for compressible flows.

Over the past 30 years, many studies have applied the ACM in conjunction with different upwind differencing schemes and solution techniques to solve steady-state as well as unsteady incompressible flow problems. The upwind differencing schemes that have been used include the flux-differencing splitting [9, 10], MUSCL [11, 12], TVD [13, 14], and WENO [15] schemes; the solution techniques that have been implemented cover the Beam–Warming scheme [16, 17], the approximation factorization–planar symmetric Gauss–Seidel scheme [13], lower–upper symmetric Gauss–Seidel (LU–SGS) scheme [15, 18], line and point relaxation schemes [10, 19], the explicit multistage Runge–Kutta method [20], and so on. The application is also extended to unsteady flow computations [9, 10, 20] by using a dual-time-stepping procedure that subiterates at each physical time step and drives the divergence of the velocity toward zero. However, the convergence rate slows down when the system gets larger or the grid becomes more stretched, which is inherent in any of the developed schemes listed above. One approach to overcoming the slow convergence is to use the preconditioning method [21, 22]; another approach is to use the multigrid method. The MG method is an efficient and generic acceleration technique known today, and it is the most efficient solver for elliptic partial differential equations [23, 24]. Numerous studies have reported successful implementation of the MG method in compressible [25] and incompressible flow problems [26–28], and its performance to accelerate convergence is very impressive. However, there are only a few studies [29–32] that developed MG methods in conjunction with the ACM, and all are exclusively based on the cell-centered finite volume discretization; see the brief review by Drikakis *et al.* [32]. Applications to unsteady

flow computations and implementations in conjunction with a collocated finite difference discretization are limited.

In this paper, a number of different implicit MG schemes in conjunction with ACM for obtaining both steady-state and unsteady solutions of the 3D INSE in general curvilinear coordinates are compared. The MG algorithm is based on the standard full approximation storage (FAS) with fixed V-cycle, except that it is now implemented in a finite difference discretization. A special feature in the implementation is the treatment of Neumann boundary conditions on the coarse grid, which received little mention in earlier literature. There are also minor differences in the restriction operators for residuals and flow variables. The solution procedure for unsteady flow is based on the dual-time-stepping procedure [10, 20]. In constructing a suitable MG smoother, it should be borne in mind that implicit schemes are advantageous over explicit ones in computing steady flows because they avoid a restrictive time-step size when highly refined grids are used to resolve viscous effects. However, there are some unsteady flow cases where the time-step size is limited by other considerations such as resolving physical time scales. The complexity in implementation and inflexibility in extending to additional PDES are more disadvantages of implicit schemes, which motivated researchers to further develop explicitly based MG techniques [33, 34]. Nevertheless, for viscous dominated flow problems, implicit schemes are more robust. This work compares three well-developed implicit schemes—the point Gauss and standard Gauss–Seidel line relaxations and the diagonalized alternating direction implicit (ADI) scheme of Beam–Warming type. We selected the first two unfactored implicit schemes mainly due to their success in compressible flows [35–37]. Relaxation and factorization schemes may have different damping abilities, which will be investigated in this work. The present point and line relaxation schemes are similar to those used by Rogers *et al.* [10, 19] but with slight modifications to the point relaxation scheme in the formation of the left-hand-side matrices and in the sweeping strategy. The author also tried the LU-SGS scheme [18, 38] and the approximation factorization-planar symmetric Gauss–Seidel scheme [13] and found that the overall computing time was greater than that of the diagonalized ADI scheme; thus these implicit schemes were excluded in this work. The discretization for the convective terms used is an upwind differencing scheme based on Roe’s approximate Riemann solver [39] with Van Leer’s MUSCL formulation [40], which was applied to the ACM and recommended for its computational efficiency and accuracy in [11, 31]. The MUSCL formulation was usually used together with limiters, but it was found that they are not necessary for the present cases. The ability of the present MG algorithm is demonstrated for two laminar flow problems, namely, the entry flow through a 90° bent square duct and the flow past a 4 : 1 prolate spheroid at incidence. Especially, the periodic flow pattern past the prolate spheroid at high incidence is simulated for the first time.

The remainder of the paper is organized as follows. In Section 2 the ACM is briefly outlined. The basic schemes are described in Section 3. The MG algorithm is presented in Section 4. Section 5 presents the results and comparison of different schemes. Finally, conclusions are drawn in Section 6.

2. GOVERNING EQUATIONS

The starting point for deriving a finite difference method is the three-dimensional, unsteady INSE written in conservative form in a generalized coordinate system (ξ, η, ζ) . The artificial compressibility is introduced after the coordinate transformation and is expressed

in terms of pseudotime derivative terms that will vanish at each physical time step in order to satisfy the continuity equation. The artificial compressibility form of the INSE is given as (cf. [10, 12])

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \mathbf{I}_m \frac{\partial \hat{\mathbf{Q}}}{\partial t} = -\frac{\partial(\hat{\mathbf{E}} - \hat{\mathbf{E}}_v)}{\partial \xi} - \frac{\partial(\hat{\mathbf{F}} - \hat{\mathbf{F}}_v)}{\partial \eta} - \frac{\partial(\hat{\mathbf{G}} - \hat{\mathbf{G}}_v)}{\partial \zeta} \equiv \hat{\mathbf{R}}, \quad (1)$$

where

$$\mathbf{I}_m = \text{diag}(0, 1, 1, 1), \quad \hat{\mathbf{Q}} = \frac{\mathbf{Q}}{J} = \frac{1}{J} \begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix}, \quad \hat{\mathbf{E}} = \frac{1}{J} \begin{bmatrix} \beta U \\ Uu + \xi_x p \\ Uv + \xi_y p \\ Uw + \xi_z p \end{bmatrix},$$

$$\hat{\mathbf{F}} = \frac{1}{J} \begin{bmatrix} \beta V \\ Vu + \eta_x p \\ Vv + \eta_y p \\ Vw + \eta_z p \end{bmatrix}, \quad \hat{\mathbf{G}} = \frac{1}{J} \begin{bmatrix} \beta W \\ Wu + \zeta_x p \\ Wv + \zeta_y p \\ Ww + \zeta_z p \end{bmatrix},$$

$$U = \xi_x u + \xi_y v + \xi_z w, \quad V = \eta_x u + \eta_y v + \eta_z w, \quad W = \zeta_x u + \zeta_y v + \zeta_z w,$$

and where β is the artificial compressibility constant, p is the static pressure, u , v , and w are the velocity components in Cartesian coordinates, U , V , and W are the contravariant velocity components in curvilinear coordinate directions, respectively, J is the Jacobian of the coordinate transformation, τ is the pseudotime, t is the physical time, $\hat{\mathbf{E}}$, $\hat{\mathbf{F}}$, and $\hat{\mathbf{G}}$ are the inviscid fluxes, and $\hat{\mathbf{E}}_v$, $\hat{\mathbf{F}}_v$, and $\hat{\mathbf{G}}_v$ are the viscous fluxes, which are given for constant viscosity as

$$\hat{\mathbf{E}}_v = \frac{\nu}{J} (\nabla \xi \cdot \nabla \xi_i) \mathbf{I}_m \frac{\partial \mathbf{Q}}{\partial \xi_i}, \quad \hat{\mathbf{F}}_v = \frac{\nu}{J} (\nabla \eta \cdot \nabla \xi_i) \mathbf{I}_m \frac{\partial \mathbf{Q}}{\partial \xi_i}, \quad \hat{\mathbf{G}}_v = \frac{\nu}{J} (\nabla \zeta \cdot \nabla \xi_i) \mathbf{I}_m \frac{\partial \mathbf{Q}}{\partial \xi_i}.$$

3. BASIC FLOW SOLVERS

The resulting hyperbolic–parabolic system (1) may be solved using existing implicit schemes. We chose the point relaxation, line relaxation, and ADI schemes in this study. These implicit schemes share the same finite difference discretization on the right-hand side (RHS) but treat the left-hand side (LHS) differently, as will be described in Sections 3.2–3.4.

3.1. Time and Spatial Discretizations

The following discussion addresses both the steady-state and the time-accurate formulations. Applying an implicit backward finite difference to the pseudotime derivatives and a second-order, three-point backward difference to the physical time derivatives, one obtains

$$\frac{\Delta \hat{\mathbf{Q}}^{n+1,m}}{\Delta \tau} = \hat{\mathbf{R}}^{n+1,m+1} + \hat{\mathbf{S}}^{n+1,m+1}, \quad (2)$$

where $\Delta \hat{\mathbf{Q}}^{n+1,m} = \hat{\mathbf{Q}}^{n+1,m+1} - \hat{\mathbf{Q}}^{n+1,m}$, $\hat{\mathbf{S}}^{n+1,m+1} = -\frac{\mathbf{I}_m}{\Delta t} (1.5 \hat{\mathbf{Q}}^{n+1,m+1} - 2 \hat{\mathbf{Q}}^n + 0.5 \hat{\mathbf{Q}}^{n-1})$, n is the physical time level, m is the pseudotime level (the number of subiterations), $\Delta \tau$ is

the pseudotime step size, and Δt is the physical time step size. After the residual terms are linearized at the $m + 1$ pseudo-time level with respect to the previous pseudotime level m , an unfactored implicit delta form is given as

$$\left[\frac{\mathbf{I}}{\Delta \tau} + \frac{1.5\mathbf{I}_m}{\Delta t} + \left(\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{Q}}} \right)^{n+1,m} \right] \Delta \hat{\mathbf{Q}}^{n+1,m} = \hat{\mathbf{R}}^{n+1,m} + \hat{\mathbf{S}}^{n+1,m}. \quad (3)$$

For the steady-state formulation, terms resulting from derivatives with respect to t are dropped in Eq. (3) and the time level $n + 1$ represents the steady state. The equations are marched in pseudotime until $\hat{\mathbf{R}}$ converges to zero. The pseudotime step $\Delta \tau$ is determined based on the CFL number. For the unsteady formulation, the equations are iterated in pseudotime so that $\hat{\mathbf{Q}}^{n+1,m+1}$ approaches the physical $\hat{\mathbf{Q}}^{n+1}$ as the RHS of Eq. (3) converges to zero. The pseudotime step $\Delta \tau$ is set to infinity, corresponding to a Newton iteration.

Both the steady-state and unsteady formulations require the discretization of the same residual vector $\hat{\mathbf{R}}$. The derivatives of the viscous fluxes in this vector are approximated using second-order central differences. However, the finite differences for the convective fluxes are critical for the quality of the numerical solutions. There are numerous ways of discretizing the convective fluxes. To be parallel with compressible flow solvers, upwind characteristic-based schemes are frequently used. In this work, Roe's flux-splitting scheme is used in conjunction with Van Leer's MUSCL formulation, which was first applied to incompressible flows by Whitfield and co-workers [11, 31]. Taking the ξ direction as an example, the inviscid derivative is discretized as

$$\frac{\partial \hat{\mathbf{E}}}{\partial \xi} \approx \tilde{\mathbf{E}}_{i+1/2} - \tilde{\mathbf{E}}_{i-1/2}, \quad (4)$$

and the numerical flux is given by

$$\begin{aligned} \tilde{\mathbf{E}}_{i+1/2} &= \hat{\mathbf{E}}_{i+1/2}(\mathbf{Q}_{i+1/2}^L) + \mathbf{A}_{i+1/2}^-(\tilde{\mathbf{Q}}_{i+1/2}) \cdot (\mathbf{Q}_{i+1/2}^R - \mathbf{Q}_{i+1/2}^L) \\ &= \hat{\mathbf{E}}_{i+1/2}(\mathbf{Q}_{i+1/2}^R) - \mathbf{A}_{i+1/2}^+(\tilde{\mathbf{Q}}_{i+1/2}) \cdot (\mathbf{Q}_{i+1/2}^R - \mathbf{Q}_{i+1/2}^L), \end{aligned} \quad (5)$$

where \mathbf{A}^\pm are matrices formed from the eigensystem of the inviscid flux Jacobian $\mathbf{A} = \partial \hat{\mathbf{E}} / \partial \mathbf{Q}$ as $\mathbf{A}^\pm = \mathbf{T}_\xi \mathbf{\Lambda}_\xi^\pm \mathbf{T}_\xi^{-1}$, $\mathbf{\Lambda}_\xi^\pm = (\mathbf{\Lambda}_\xi \pm |\mathbf{\Lambda}_\xi|) / 2$, and $\mathbf{\Lambda}_\xi$, \mathbf{T}_ξ , and \mathbf{T}_ξ^{-1} are calculated according to the formulae in Ref. [17], which are given in the Appendix. In incompressible flows, the Roe average can be replaced by a simple arithmetic average, $\tilde{\mathbf{Q}}_{i+1/2} = (\mathbf{Q}_{i+1/2}^R + \mathbf{Q}_{i+1/2}^L) / 2$. The flow variables to the left and right of a node interface are obtained from the MUSCL formulation [41]

$$\mathbf{Q}_{i+1/2}^L = \mathbf{Q}_i + \left\{ \frac{1}{4} [(1 - \chi)\nabla + (1 + \chi)\Delta] \mathbf{Q} \right\}_i, \quad (6)$$

$$\mathbf{Q}_{i+1/2}^R = \mathbf{Q}_{i+1} - \left\{ \frac{1}{4} [(1 - \chi)\Delta + (1 + \chi)\nabla] \mathbf{Q} \right\}_{i+1}, \quad (7)$$

where $\chi = 1/3$, corresponding to a third-order upwind-biased approximation, is used for the present computations. Zero-order extrapolation of ∇ and Δ is used at grid points next to the left and right boundaries, respectively.

A direct solution of Eq. (3) is not feasible for general three-dimensional problems because the resulting large block and banded matrix are very costly to invert and require a large

amount of storage. To reduce the bandwidth of the resulting LHS matrix in Eq. (3), the Jacobian of $\hat{\mathbf{R}}$ in the LHS is formed using a residual based on the first-order form of (5). This results in the diagonally dominant form

$$\begin{aligned}
 & \left[\frac{1}{J} \left(\frac{\mathbf{I}}{\Delta\tau} + \frac{1.5\mathbf{I}_m}{\Delta t} \right) + (\mathbf{A}^+ + \mathbf{A}_v)_{i-1/2} - (\mathbf{A}^- - \mathbf{A}_v)_{i+1/2} + (\mathbf{B}^+ + \mathbf{B}_v)_{j-1/2} \right. \\
 & \quad \left. - (\mathbf{B}^- - \mathbf{B}_v)_{j+1/2} + (\mathbf{C}^+ + \mathbf{C}_v)_{k-1/2} - (\mathbf{C}^- - \mathbf{C}_v)_{k+1/2} \right] \Delta\mathbf{Q}_{i,j,k}^{n+1,m} \\
 & \quad - (\mathbf{A}^+ + \mathbf{A}_v)_{i-1/2} \Delta\mathbf{Q}_{i-1,j,k}^{n+1,m} + (\mathbf{A}^- - \mathbf{A}_v)_{i+1/2} \Delta\mathbf{Q}_{i+1,j,k}^{n+1,m} \\
 & \quad - (\mathbf{B}^+ + \mathbf{B}_v)_{j-1/2} \Delta\mathbf{Q}_{i,j-1,k}^{n+1,m} + (\mathbf{B}^- - \mathbf{B}_v)_{j+1/2} \Delta\mathbf{Q}_{i,j+1,k}^{n+1,m} \\
 & \quad - (\mathbf{C}^+ + \mathbf{C}_v)_{k-1/2} \Delta\mathbf{Q}_{i,j,k-1}^{n+1,m} + (\mathbf{C}^- - \mathbf{C}_v)_{k+1/2} \Delta\mathbf{Q}_{i,j,k+1}^{n+1,m} \\
 & = (\hat{\mathbf{R}} + \hat{\mathbf{S}})^{n+1,m}, \tag{8}
 \end{aligned}$$

where the viscous Jacobians are formed by dropping the off-diagonal elements,

$$\mathbf{A}_v = \frac{\nu}{J} (\nabla\xi \cdot \nabla\xi) \mathbf{I}_m, \quad \mathbf{B}_v = \frac{\nu}{J} (\nabla\eta \cdot \nabla\eta) \mathbf{I}_m, \quad \mathbf{C}_v = \frac{\nu}{J} (\nabla\zeta \cdot \nabla\zeta) \mathbf{I}_m. \tag{9}$$

In practice, one always resorts to relaxation or factorization schemes. It is generally known that the factorization error of factorized schemes limits the size of the time step and factorized schemes need fewer operations per time step, while larger time steps can be used for relaxation schemes. Therefore, it is worthwhile to compare some of the widely used schemes for future development of more efficient methods. We will briefly outline a modified point Gauss relaxation scheme, a standard line Gauss–Seidel relaxation scheme, and a diagonalized Beam–Warming scheme for Eq. (8).

3.2. Point Relaxation Scheme

The point relaxation scheme used is a Gauss point relaxation that iteratively solves a block diagonal system formed by multiplying all of the nonmain block diagonals in the LHS by the most recently calculated $\Delta\mathbf{Q}$ and moving to the RHS in Eq. (8). This is done for each interior point, sweeping sequentially through the computational domain. A forward sweep plus a backward sweep counts as one sweep, denoted as PR(1). In the current implementation, it is found that PR(2) is optimum in all PR(n). A slight modification to the point relaxation in [19] is as follows: (i) Every PR(2) is initialized by setting all $\Delta\mathbf{Q}$ to zero and (ii) all the Jacobians are updated following the updating of \mathbf{Q} after every PR(2) and are the same as those for calculating the RHS by Eq. (5).

3.3. Line Relaxation Scheme

A standard Gauss–Seidel line relaxation scheme similar to [10] is used. A solving direction is chosen such that terms on the LHS from points along this direction are solved for simultaneously. All terms on the LHS from points off this direction are multiplied by the most recently calculated $\Delta\mathbf{Q}$ and moved to the RHS. The resulting block tridiagonal systems are solved for each grid line as the domain is swept forward and backward. A forward sweep plus a backward sweep again counts as one sweep, denoted as LR(1). In

the current implementation, it is found that LR(1) is optimum. Every LR(1) is also initialized by setting $\Delta\mathbf{Q}$ to zero. All the Jacobians are updated after every LR(1). The three coordinate directions are alternately chosen as the solving direction when the pseudotime goes on.

3.4. Diagonalized ADI Scheme

The diagonal ADI scheme as developed by Pulliam *et al.* [42] and applied to ACM by Rogers *et al.* [17] is extended here as

$$\begin{aligned} & \mathbf{T}_\xi[\mathbf{D} + \delta_\xi^- \mathbf{\Lambda}_\xi^+ + \delta_\xi^+ \mathbf{\Lambda}_\xi^- - \delta_\xi \mathbf{A}'_v \delta_\xi] \mathbf{T}_\xi^{-1} \mathbf{D}^{-1} \mathbf{T}_\eta[\mathbf{D} + \delta_\eta^- \mathbf{\Lambda}_\eta^+ + \delta_\eta^+ \mathbf{\Lambda}_\eta^- - \delta_\eta \mathbf{B}'_v \delta_\eta] \mathbf{T}_\eta^{-1} \\ & \times \mathbf{D}^{-1} \mathbf{T}_\zeta[\mathbf{D} + \delta_\zeta^- \mathbf{\Lambda}_\zeta^+ + \delta_\zeta^+ \mathbf{\Lambda}_\zeta^- - \delta_\zeta \mathbf{C}'_v \delta_\zeta] \mathbf{T}_\zeta^{-1} \Delta\mathbf{Q}^{n+1,m} = \hat{\mathbf{R}}^{n+1,m} + \hat{\mathbf{S}}^{n+1,m}, \quad (10) \end{aligned}$$

where $\mathbf{D} = \frac{1}{J}(\frac{1}{\Delta\tau} + \frac{1.5}{\Delta t})\mathbf{I}$, δ_ξ^- and δ_ξ^+ are two-point backward and forward differences, δ_ξ is the central difference for viscous terms, and $\mathbf{A}'_v = \frac{v}{J}(\nabla\xi \cdot \nabla\xi)\mathbf{I}$ is formed from Eq. (9) by changing \mathbf{I}_m to identity matrix \mathbf{I} .

3.5. Boundary Conditions

The present study involves both external and internal flows. For the external flow past an isolated prolate spheroid, we use a special grid layout where the first and last $\xi = const$ grid planes are located at $\frac{1}{2}\Delta\xi$ away from the polar axes of the prolate spheroid, and we solve for the flow variables in the two planes in the same way as for other inner points. In doing so, $\hat{\mathbf{E}}_{i-1/2}$ and $\hat{\mathbf{E}}_{v_{i-1/2}}$ for the points in the first ξ plane, or $\hat{\mathbf{E}}_{i+1/2}$ and $\hat{\mathbf{E}}_{v_{i+1/2}}$ for the points in the past ξ plane, are set to zero. Reflection conditions are also used when (6) and (7) are evaluated. This technique avoids the polar singularity in a way similar to that adopted in a cell-centered finite volume discretization. Although there are other ways to treat the polar singularity in a finite difference method, they are either too crude, such as the one that took the flow variables on the polar axis as averaging from around the axis, or too complicated, such as the one where equation transformation is made at the polar axis [43]. The flow variables in the farfield are determined through a characteristic analysis normal to the boundary [16]. The nonslip condition for the velocity is specified on the wall surface and the pressure is obtained by applying zero normal pressure gradient. Symmetric or periodic conditions are imposed in the symmetry plane. For the internal flow through a square duct, the inflow velocity profile is prescribed. The outflow boundary condition is a prescribed pressure and a zero normal velocity gradient.

The boundary conditions are generally applied implicitly with the two relaxation schemes, where $\Delta\mathbf{Q}$ has the same physical meaning as the original flow variables. However, where there are implementation difficulties, such as on the characteristics boundary, the boundaries are treated explicitly (setting $\Delta\mathbf{Q}$ on the boundary to zero while solving for $\Delta\mathbf{Q}$ in the interior points). As with the diagonalized ADI scheme, all boundaries are treated explicitly because of the lack of physical meaning for $\Delta\mathbf{Q}$ at split steps.

4. MULTIGRID METHOD

The MG method adopted in this work is a full-coarsening nonlinear FAS V-cycle algorithm implemented with a finite difference discretization. The fine grid in the MG method

serves to dampen the high-frequency errors, while the coarser grid dampens the low-frequency errors. The finest grid G_1 , which describes the shape most accurately, is set up first. A sequence of coarse grids G_2, G_3, G_N can be formed by successively deleting every other line in all three coordinate directions on the previous finer grid. The solution starts from the finest grid (top level) and goes to the coarsest grid (lowest level) for restriction, and the prolongation is performed from the lowest level upward to the top level. For illustration, consider a typical numerical scheme

$$\mathcal{M}^k \Delta \mathbf{Q}^k = \mathcal{R}(\mathbf{Q}^k), \quad (11)$$

where \mathcal{M}^k is the LHS implicit operator of Eq. (8) or (10), \mathcal{R} is the residual $\hat{\mathbf{R}} + \hat{\mathbf{S}}$, and the superscript k is the grid level. One FAS V-cycle for Eq. (11) consists of the following steps:

Step 1 (fine grid solution). Carry out one iteration of Eq. (11) on the finest grid ($k = 1$) to obtain an approximate solution \mathbf{Q}^1 .

Step 2 (coarse grid solution). If the present grid is the coarsest, go to Step 3. Otherwise, restrict the solution and updated residuals to the next coarser grid by

$$\mathbf{Q}_0^{k+1} = I_k^{k+1} \mathbf{Q}^k, \quad \mathbf{R}_c^{k+1} = \bar{I}_k^{k+1} \mathbf{R}^k, \quad (12)$$

where the subscripts 0 and c denote the initialized coarse grid solution and the transferred residuals, respectively, and I_k^{k+1} and \bar{I}_k^{k+1} are corresponding restriction operators. Then a coarse grid forcing function, which represents the relative truncation error between the coarse and the fine grid, is defined as [25]

$$\mathbf{P}^{k+1} = \mathbf{R}_c^{k+1} - \mathcal{R}(\mathbf{Q}_0^{k+1}). \quad (13)$$

The residual on the coarse grid is calculated as

$$\mathbf{R}^{k+1} = \mathcal{R}(\mathbf{Q}^{k+1}) + \mathbf{P}^{k+1}, \quad (14)$$

and the coarse grid equation is given by

$$\mathcal{M}^{k+1} \Delta \mathbf{Q}^{k+1} = \mathbf{R}^{k+1}. \quad (15)$$

Perform one iteration at intermediate grid levels and five iterations at the coarsest grid level. Repeat Step 2 until the coarsest grid is reached.

Step 3 (coarse grid correction and postrelaxation). If the coarsest grid is reached, the obtained coarse grid corrections are interpolated back to the next finer grid and the old solution is updated by

$$\mathbf{Q}_{\text{new}}^k = \mathbf{Q}^k + P_{k+1}^k (\mathbf{Q}^{k+1} - \mathbf{Q}_0^{k+1}), \quad (16)$$

where P_{k+1}^k is the prolongation operator. Use $\mathbf{Q}_{\text{new}}^k$ as an initial guess and iterate Eq. (15) one time on grid G_k to obtain an approximate solution. Repeat Step 3 until the finest grid is reached.

In accordance with a finite difference discretization, the restriction for the solution is a simple injection due to the collocation of coarse and fine grid points, while the restriction for the residuals is a full-weighted average over all the fine grid points composing the

control volume of each coarse grid point. Standard trilinear interpolation is used for the prolongation operator. In addition, a proper treatment of Neumann boundary conditions on the coarse grid is found to be essential in iterating Eq. (15). Basically, there are two ways to update the flow variables on the Neumann-type boundaries. One is to directly and explicitly update the boundary flow variable itself from the already updated interior values by

$$\mathbf{Q}_b^{s+1} = f(\mathbf{Q}_{\text{interior}}^{s+1}); \quad (17)$$

another is to update the flow variable in the incremental form, either explicitly by

$$\mathbf{Q}_b^{s+1} = \mathbf{Q}_b^s + \Delta \bar{\mathbf{Q}}_b^s, \quad (18)$$

or implicitly by

$$\mathbf{Q}_b^{s+1} = \mathbf{Q}_b^s + \Delta \mathbf{Q}_b^s, \quad (19)$$

where the superscript s is the iteration index of the coarse grid equation, subscript b represents boundary, $\Delta \bar{\mathbf{Q}}_b^s$ is explicitly obtained from linearization of the Neumann boundary condition, and $\Delta \mathbf{Q}_b^s$ is implicitly solved for from the interior point equations coupled with the linearized condition. As for the computational effects of different treatments, it will be shown in the numerical example that direct updating (17) will lead to stalling of the residuals or divergence in some cases, and explicit incremental updating (18) will lead to slow convergence. Only the proposed implicit incremental form (19) will ensure good convergence.

5. RESULTS AND DISCUSSION

The present MG algorithm in conjunction with each of the three implicit schemes is applied to two 3D laminar flow problems: the flow through a square duct with a 90° bend and the flow past a prolate spheroid with long/short axis ratio of 4 : 1. The unsteady flow past the prolate spheroid at higher incidence is also simulated. The choice of the artificial compressibility factor β is important [10, 16]. We found that $\beta = 1$ gives optimum convergence for all schemes used. The convergence of the ADI scheme is more sensitive to β than the convergence of the relaxation schemes, and the range of β for the ADI scheme to converge is quite narrow. However, the relaxation schemes will converge well for β in the range 1–10. Thus we set β to 1 for the ADI scheme and 10 for the relaxation schemes. For steady-state flows, the pseudotime step is spatially varied based on an optimal CFL number arrived at through convergence tests. The optimal CFL number is around 2.5–10 for the ADI scheme and 100–600 for the relaxation schemes. The three-space dimensionality and the inherent factorization error of the ADI scheme limit its CFL number to values smaller than those of the relaxation schemes. For unsteady flows, the pseudotime step is set to infinity, while the constant physical time step is adopted on all grid levels. The initial flowfield is a uniform free-stream condition in the prolate spheroid problem or a prescribed inflow velocity profile and zero pressure in the square duct problem.

5.1. Steady Flow Through a 90° Bending Square Duct

The experiment of Humphrey *et al.* [44], which measured the flow through a strongly curved 90° bend square duct, was used as a steady-state test case in the present study. The

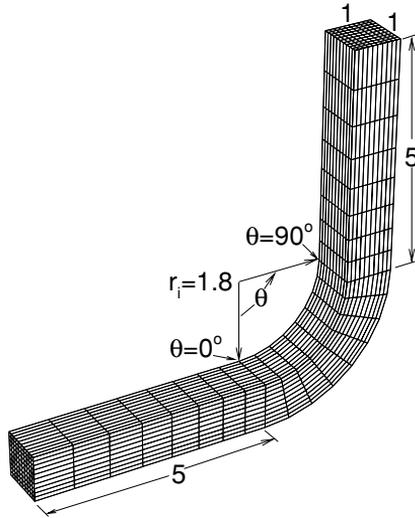


FIG. 1. Geometry and grid ($25 \times 11 \times 11$) of flow through a 90° bending square duct.

problem was nondimensionalized using the side of the square cross section as the unit length and the average inflow velocity as the unit velocity. The Reynolds number was 790 based on the unit length and velocity. The geometry and an example grid system of $25 \times 11 \times 11$ are shown in Fig. 1. The straight inflow section before the bend was set to a length of 5, and the outflow section downstream of the bend was also set to a length of 5. The radius of curvature of the inner wall in the curved section was 1.8 while that of the outer wall was 2.8. The inflow velocity profile was prescribed to be that of a fully developed laminar straight square duct as given by White [45].

In the present work, three different grid systems with grid points of $49 \times 33 \times 33$, $73 \times 49 \times 49$, and $121 \times 61 \times 61$ were used to solve the problem. As an example of how the convergence behaves for different boundary condition treatments on the coarse grid, the convergence history for the $49 \times 33 \times 33$ grid is shown in Fig. 2, where the averaged L2-norm of residuals is plotted vs iteration number. It can be seen that the fastest convergence was obtained by using the proposed coarse grid boundary correction Eq. (19). Updating the boundary values in explicit incremental form will lead to slower convergence (Eq. (18)), while direct application of the Neumann condition to \mathbf{Q} itself will result in stalling of the residuals (Eq. (17)). Figure 3 shows the comparison of the convergence history between the MG and single grid iterations, where the work unit is equivalent to the iteration number of the single grid ADI scheme, and symbols are marked on the plotting lines for relaxation MG schemes at every 50 iterations. In general, the CPU time of the MG solution is only 1/3 to 1/4 that of its single grid counterpart. Nevertheless, the CPU time of the PR(2) MG scheme is about 80% that of the line relaxation MG scheme and is less than 50% that of the ADI MG scheme. Thus it is shown that the PR(2) or the standard line relaxation scheme is an efficient MG smoother in three space dimensions, but the ADI scheme is not as effective as the other two schemes for this test problem.

A grid-independence study using the three grid systems demonstrated that the $73 \times 49 \times 49$ grid and above can give grid-independent results. Figure 4 shows the computed results in comparison with the experimental data of Humphrey *et al.* [44]. The x axis is the normalized radial distance and the y axis is the velocity in the streamwise direction. The streamwise

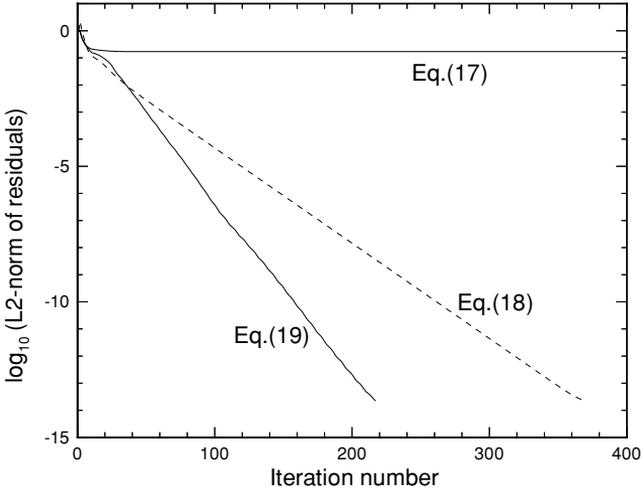


FIG. 2. Comparison of the convergence histories among different boundary condition treatments on the $49 \times 33 \times 33$ grid for the flow through the 90° bending square duct. The computations were done using the three-level PR(2) multigrid scheme.

velocity (V_θ) profile at the x - y symmetry plane is shown for four different streamwise stations along the duct. The four positions correspond to the cross-flow planes at θ equal to 0° , 30° , 60° , and 90° in the curved section (see Fig. 1). The match between computation and experiment is fairly good at the first two stations. However, discrepancy becomes evident between the computed and experimental results when the flow begins to form swirls at $\theta = 60^\circ$ station. This deviation can also be found in other numerical calculations [10, 15] and is thought to be caused by any small change in the Reynolds number or small amount of turbulent mixing [10]. Nevertheless, the peak of streamwise velocity near the outside wall and the second peak near the inside wall at $\theta = 60^\circ$ and 90° are well captured.

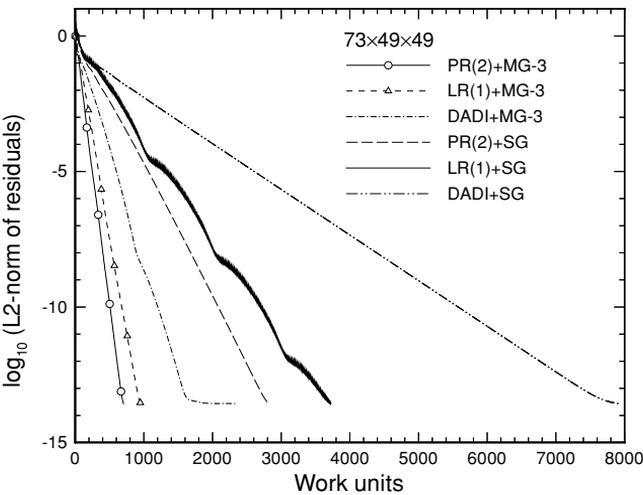


FIG. 3. The convergence histories of the three-level multigrid (MG-3) and single grid (SG) computations in conjunction with the three implicit schemes. The symbols are marked on two plotting lines every 50 iterations.

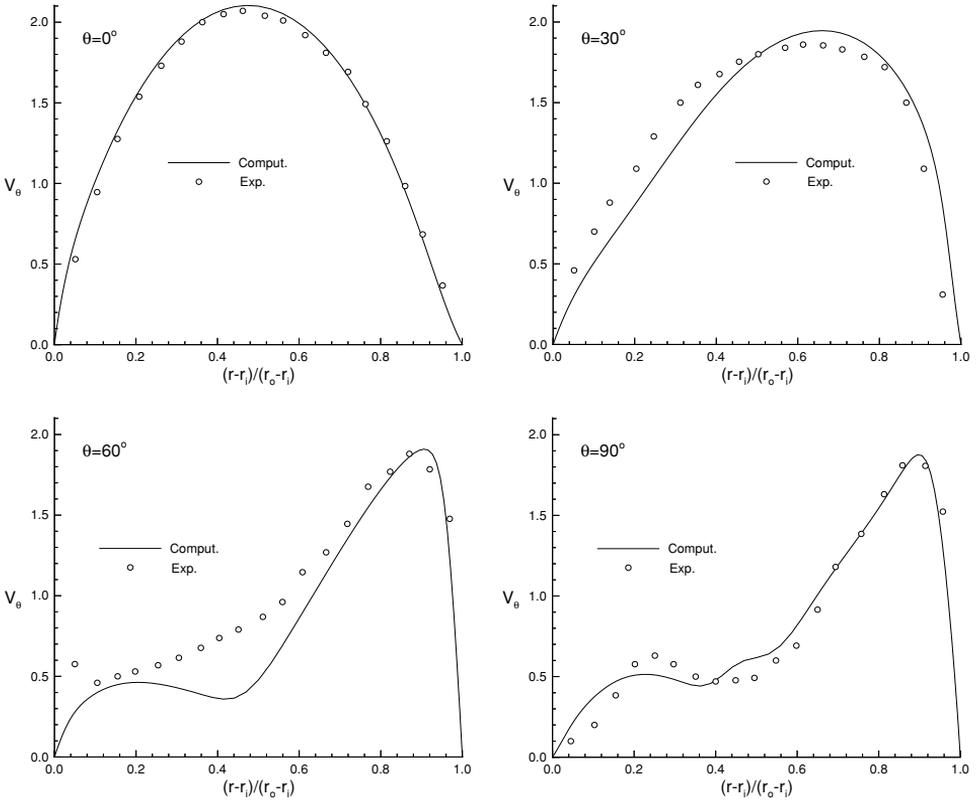


FIG. 4. Comparison of computed streamwise velocity profiles at four streamwise locations (midspan) with the experimental results [44].

5.2. Steady Flow Past a 4 : 1 Prolate Spheroid

The flow past a prolate spheroid has been extensively studied for more than three decades and is a benchmark problem for studying three-dimensional separated flow structures [46, 47]. Two axis ratios of the prolate spheroid, namely, 6 : 1 and 4 : 1, were commonly investigated in past studies. For a 6 : 1 prolate spheroid, detailed surface shear stress, pressure, and oil flow pattern measurements have been performed by Kreplin *et al.* [48], which provided a data set for validating the ability of turbulence modeling of Navier–Stokes solvers [49, 50]. For a 4 : 1 spheroid, there are more flow visualization studies [46, 47] than quantitative measurements. However, the vortical flow characteristics on a 4 : 1 spheroid are much richer than those on a 6 : 1 spheroid. For this reason, the laminar flow past a 4 : 1 prolate spheroid is studied in this paper.

An O–O grid topology is used for the computations. A representative O–O type grid system of $19(\xi) \times 17(\eta) \times 25(\zeta)$ is shown in Fig. 5 for the bare prolate spheroid, where ξ is in the streamwise direction, η in the circumferential direction, and ζ in the normal direction. The grids are generated by using the ellipsoid coordinates and are clustered in the normal direction near the body surface for resolving the thin viscous layers. The minimum grid spacing in the normal direction normalized with the major axis is taken to be 1.0×10^{-4} . The first and the last $\xi = \text{const}$ planes are located at $\frac{1}{2}\Delta\xi$ away from the axes to avoid the

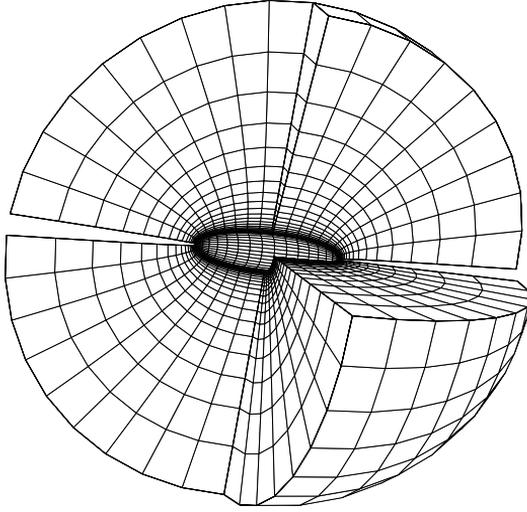
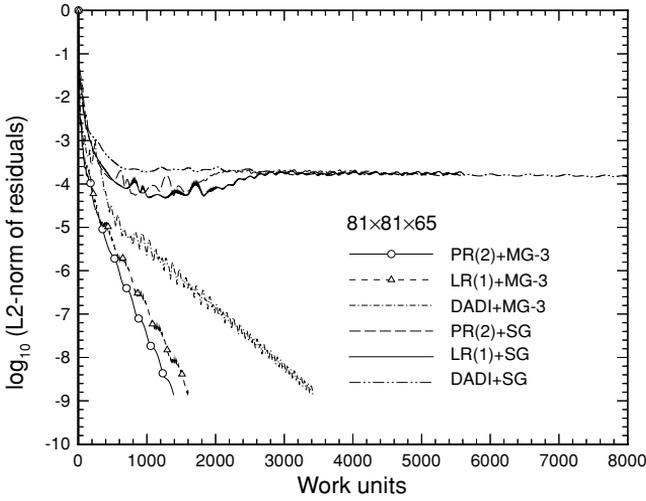


FIG. 5. A sketch grid of $19 \times 17 \times 25$ with the first and last ξ grid located at half grid size away from the singular axes for computation of flow past a 4 : 1 prolate spheroid.

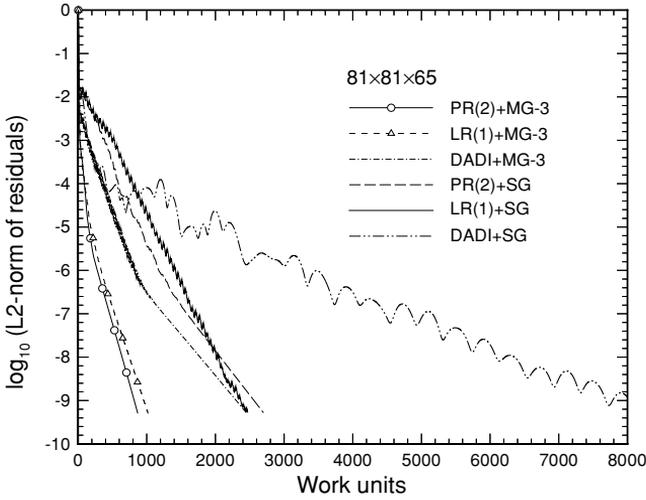
polar singularity. The grid extends 1.5 times the length upstream and downstream. To show how results depend on the grid size, calculations have been performed on a coarse grid with $37 \times 33 \times 49$ points and on a fine grid with $81 \times 81 \times 65$ points.

Generally, steady-state flows exist at low Reynolds number and low incidence, while unsteady flows appear at high Reynolds number and/or high incidence. Two steady-state cases were studied: one is for a low incidence ($\alpha = 10^\circ$) and a moderate Reynolds number ($Re = \frac{U_\infty L}{\nu} = 11700$) that is close to the onset of flow unsteadiness; another is for a higher incidence ($\alpha = 30^\circ$) and a lower Reynolds number ($Re = 3000$). Figure 6 compares the convergence histories between the MG and single grid solutions and among different implicit schemes on the fine grid system. It can be seen that steady-state flow can be obtained in both cases by using the MG schemes, but only a nominally steady flow can be obtained for the first case by using the single grid schemes because the residual cannot be reduced below a certain level. It can also be seen from Fig. 6b that the computational work required by the MG schemes is only 30–45% that required by their single grid counterparts. The convergence acceleration is not as good as those in the square duct case. However, in view of the extremely long computing time required in an implicit scheme, the observed reduction in CPU time is quite significant. Both the modified point relaxation (PR(2)) and the standard line relaxation MG schemes outperform the diagonalized ADI MG scheme by saving over 50% in CPU time. The CPU time of the PR(2) + MG scheme is slightly less than that of the LR(1) + MG scheme.

To compare the computational resolution of the separated flows for different grid sizes, the surface streamline patterns are shown in Fig. 7. It is evident that surface streamline patterns as computed using the two grid systems are in close agreement with each other in front of the main separation line but differ in the size and flow topology of the separated region. In the case of $\alpha = 10^\circ$, the separated region moves upstream on the fine grid, and an attachment node N_a and saddle point S_2 appear. There exists mainly so-called closed-type separation [46, 47]. In the case of $\alpha = 30^\circ$, the separation node N_{s1} appears



(a) $\alpha = 10^\circ$, $Re = 11700$



(b) $\alpha = 30^\circ$, $Re = 3000$

FIG. 6. The convergence histories of the multigrid and single grid computations in conjunction with the three implicit schemes for flow past a 4 : 1 prolate spheroid. The symbols are marked on two plotting lines at every 50 iterations.

on the fine grid. Both open- and closed-type separations are evident. The surface pressure distributions for the windward ($\theta = 0^\circ$) and leeward ($\theta = 180^\circ$) planes of symmetry are given in Fig. 8. The agreement between the fine and coarse grids is good in both cases.

5.3. Unsteady Flow Past a 4 : 1 Prolate Spheroid at 50° Incidence

In the previous steady-state calculations, it was shown that the MG algorithm combined with each of the three implicit schemes provides significant convergence acceleration and that the point relaxation (PR(2)) scheme is more efficient than the other two schemes. Therefore, only the point relaxation scheme and its MG algorithm are compared in the computation of the unsteady flow past the prolate spheroid at higher incidence.

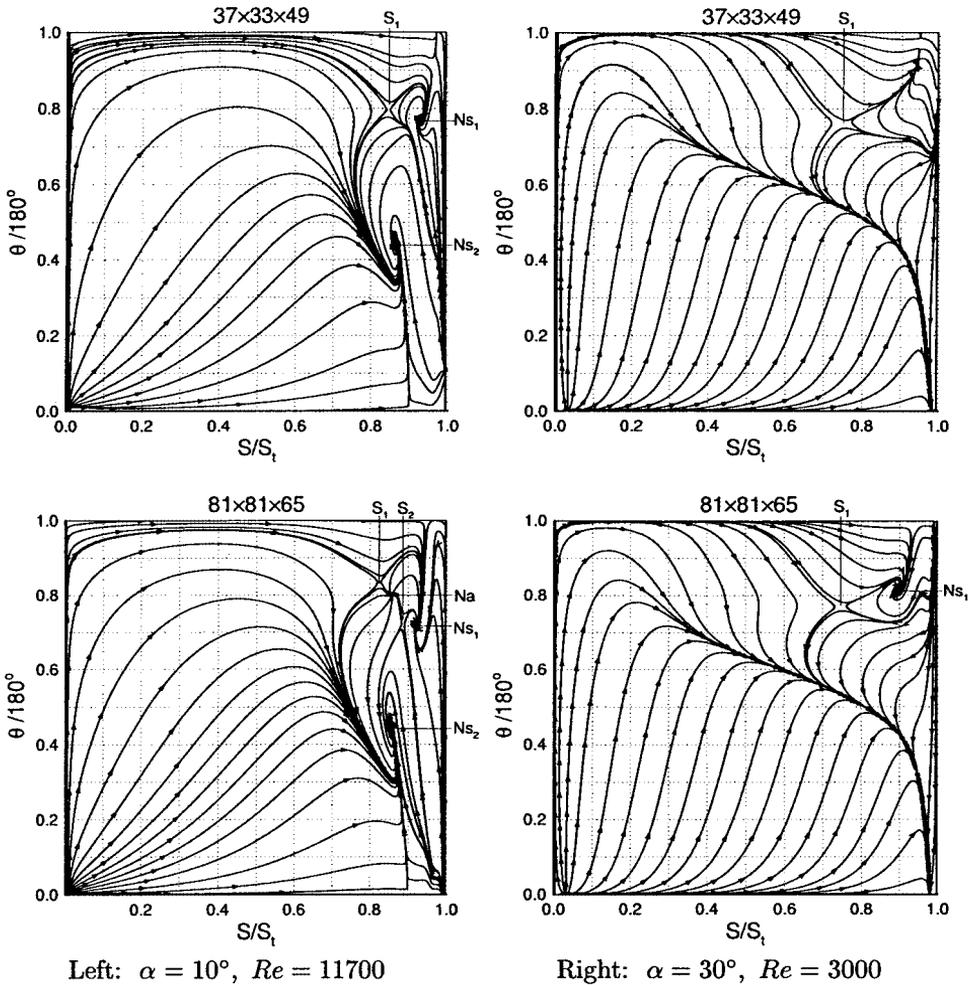


FIG. 7. Comparison of the surface streamlines on the unwrapped spheroid surface as computed with different grid sizes. S is the surface arclength in the longitudinal direction, S_1 is the total arclength, $\theta = 0$ is the windward symmetry, and $\theta = 180^\circ$ is the leeward symmetry.

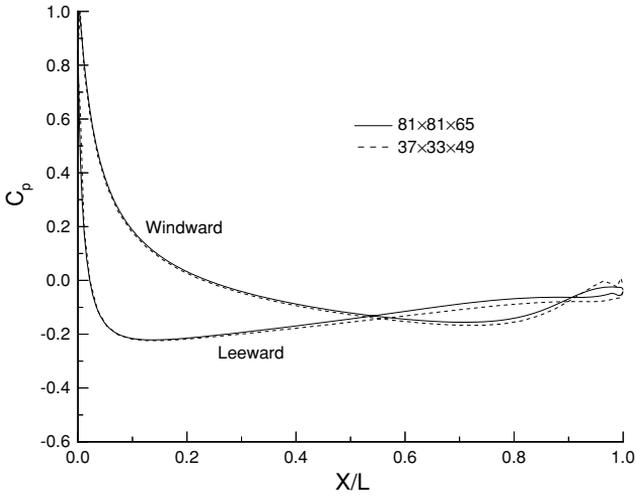
To simulate the unsteady flow, a full flowfield around the body was used. The computations were done on a grid system of $57 \times 97 \times 57$. To find out quickly when the flow becomes unsteady, several runs using pseudotime marching only were performed first. The Reynolds number was fixed at 3000 while the incidence was continually increased from 30° . We found that the flow became periodic first at $\alpha = 43^\circ$. Then a series of runs were conducted for $\alpha = 50^\circ$ using the dual-time stepping method implemented with both single grid and MG schemes. The effects of different values of ε for the residuals reduction in subiterations were compared to establish an appropriate convergence criterion for a time-accurate solution. The computations used a physical time step $\Delta t = 0.05$ and were run for two periods of the side force coefficient.

For each case, Table I reports the CPU time on a Pentium III-550E computer, the number of subiterations used, the side force and normal force coefficients, and the Strouhal number based on the period of the side force T . These results show that the problem only requires

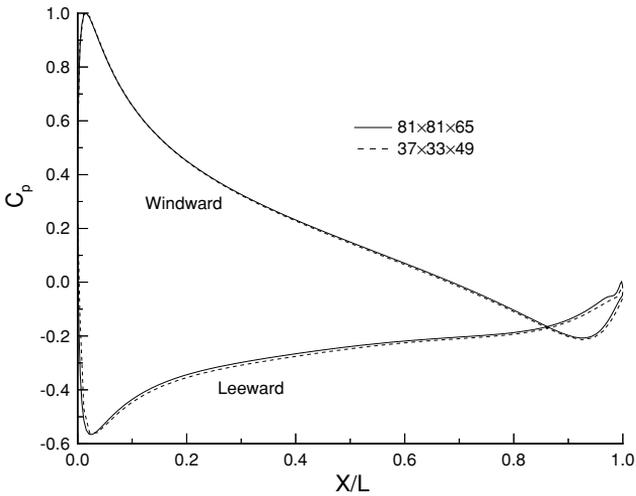
TABLE I
Quantities for the Spheroid at $\alpha = 50^\circ$, $Re = 3000$, $57 \times 97 \times 57$ Grid Points

ε	CPU hour (MG/SG)	Subiterations (MG/SG)	Side force ^a (C_s)	Normal force ^a (C_n)	Strouhal no. ($L \sin \alpha / U_\infty T$)
10^{-1}	18.1/32.8	5-8/13-20	± 0.05932	0.5272 ± 0.0074	0.04974
10^{-2}	40.6/57.8	12-16/25-31	± 0.05933	0.5272 ± 0.0075	0.04975
10^{-3}	69.7/87.9	20-29/40-55	± 0.05933	0.5272 ± 0.0075	0.04975

^a The force coefficient is based on the projection area $S = \pi ab$ of the prolate spheroid, where a and b are the half-long and half-short axes, respectively.



(a) $\alpha = 10^\circ$, $Re = 11700$



(b) $\alpha = 30^\circ$, $Re = 3000$

FIG. 8. Comparisons of the surface pressure distributions at the windward ($\theta = 0^\circ$) and leeward ($\theta = 180^\circ$) symmetry planes.

the residual to drop one order of magnitude ($\varepsilon = 10^{-1}$) from its initial value at each physical time step to obtain adequate accuracy. The value $\varepsilon = 0.1$ requires only 5–8 subiterations per time step for the MG solution and 13–20 subiterations for the single grid solution. The result supports an earlier statement [10] that any work done beyond 5 subiterations did not change the solution of the unsteady flow computations. At $\varepsilon = 0.1$, the CPU time of the MG scheme is 55% that of the single grid scheme. However, at further enhanced convergence levels, the CPU time of the MG scheme is 70–80% that of the single grid scheme. Actually, the corresponding subiterations also depend on the size of the physical time step and complexity of the flow problem in addition to the convergence level. In view of the fact that $\varepsilon = 0.1$ results in satisfactory accuracy within a few subiterations and that $\varepsilon = 10^{-4}$ is only necessary in occasional cases such as the brief stage started impulsively from free-stream conditions, we think that the PR(2) + MG scheme implemented with the dual-time stepping method is plausible for calculating unsteady incompressible flows. Perhaps it is possible to improve the efficiency of the MG scheme by using other techniques such as preconditioning [34, 51].

Figure 9 shows the time history of the normal force and side force coefficients. The flow is fully periodic. The calculated period of the side force is $15.4 L/U_\infty$. Figure 10 shows the time development of the total pressure contour in the cross-section plane normal to the major axis within one period of the side force, where the time instants correspond to t_1 through t_9 as marked in Fig. 9. It can be seen that alternate vortex shedding happens near the body surface in a way similar to that observed in a 2D flow past a cylinder. But the shed vortex is now convected downstream in a direction almost perpendicular to the cross-section plane, not in the transverse direction in this plane, which would form a wake. The flow at t_1 is the same as that in t_9 , and both are reflectively symmetric with respect to the longitudinal symmetry plane with the flow at t_5 ; so are the flows in t_2 and t_6 , t_3 and t_7 , and t_4 and t_8 .

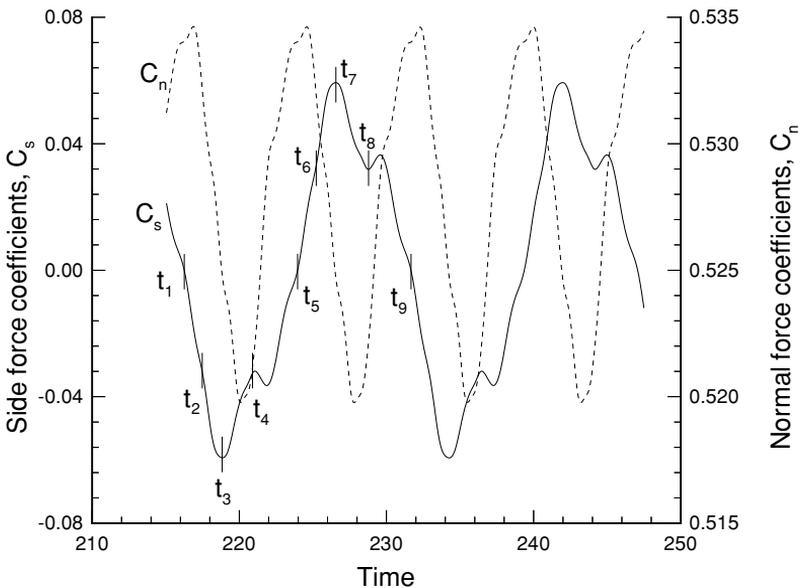


FIG. 9. Time history of the side force and normal force coefficients for flow past a 4:1 prolate spheroid at $\alpha = 50^\circ$, $Re = 3000$.

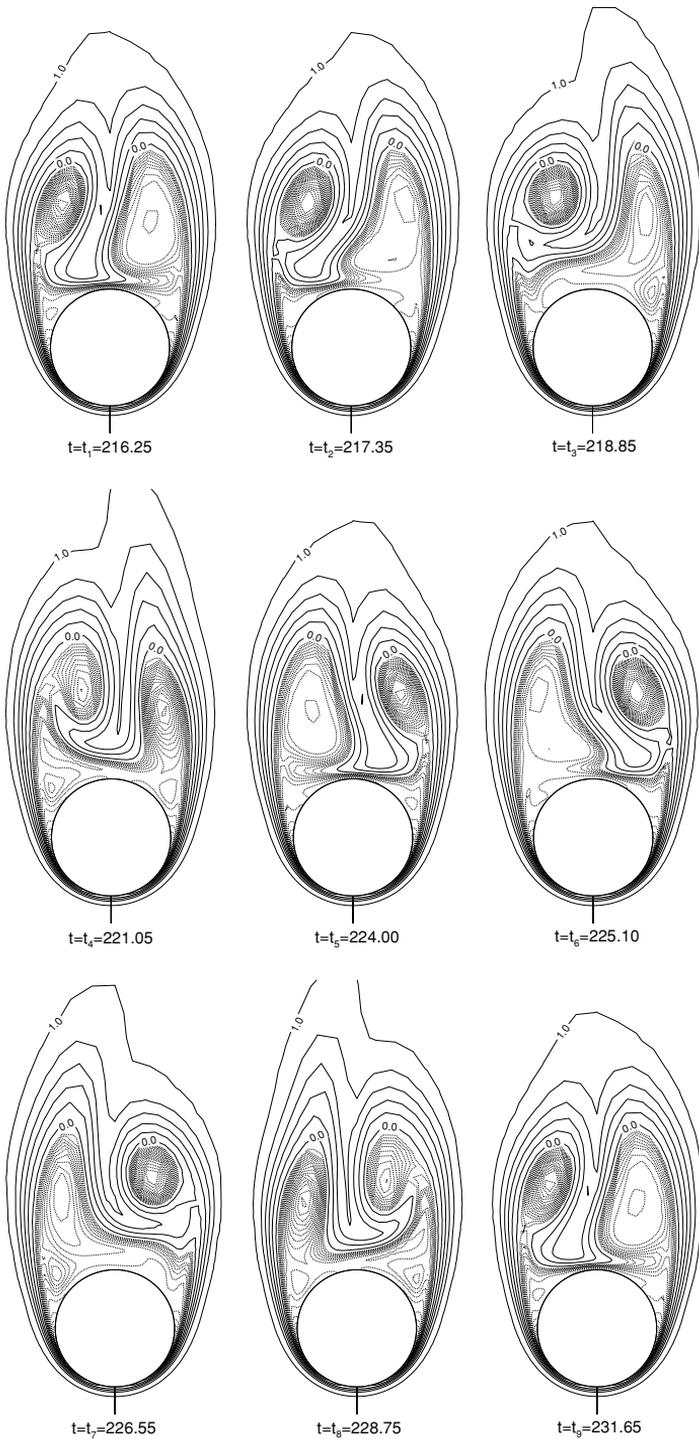


FIG. 10. Time development of the total pressure contours in the cross-section plane at $X/L = 0.75$ for flow past a 4 : 1 prolate spheroid, $\alpha = 50^\circ$, $Re = 3000$. The positive values are plotted as solid lines with an increment $\Delta P_0 = 0.2$, while the negative values are plotted as dotted lines with $\Delta P_0 = 0.025$.

6. CONCLUSIONS

A nonlinear FAS multigrid algorithm in conjunction with three frequently used implicit schemes has been implemented and applied to simulations of both steady-state and unsteady three-dimensional incompressible laminar flows. An appropriate treatment of Neumann boundary conditions on coarse grids is proposed for the finite difference discretization, which is shown to guarantee good convergence of the resulting multigrid schemes. The numerical tests show that the multigrid scheme implemented with the modified point Gauss relaxation (PR(2)) or the standard line Gauss relaxation is more efficient than that implemented with the ADI scheme in three-dimensional flows, with the PR(2) + MG scheme being most efficient. It is demonstrated that all the multigrid schemes accelerate the convergence by at least two times their corresponding single grid schemes in computing steady-state flows. However, the convergence acceleration of the multigrid algorithm for unsteady flows is relatively small. It is found that a one order of magnitude reduction of the residuals in the subiteration can meet the time-accurate requirement in general, except in a brief sharp transitional stage. The separated flow structures as computed using the third-order MUSCL scheme are fairly good when a fine grid is used. The simulated periodic flow past a 4 : 1 prolate spheroid at higher incidence also indicates that the modified point relaxation (PR(2)) multigrid scheme in conjunction with the ACM is preferable when large-scale computations of unsteady three-dimensional incompressible flows are conducted.

APPENDIX: INVISCID JACOBIAN

The Jacobian matrices \mathbf{A} , \mathbf{B} , \mathbf{C} ($\mathbf{A} = \frac{\partial \hat{\mathbf{E}}}{\partial \mathbf{Q}}$, $\mathbf{B} = \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{Q}}$, $\mathbf{C} = \frac{\partial \hat{\mathbf{G}}}{\partial \mathbf{Q}}$), are given by

$$\mathbf{A}_i = \frac{1}{J} \begin{bmatrix} 0 & k_x \beta & k_y \beta & k_z \beta \\ k_x & k_x u + \Theta & k_y u & k_z u \\ k_y & k_x v & k_y v + \Theta & k_z v \\ k_z & k_x w & k_y w & k_z v + \Theta \end{bmatrix}, \quad (\text{A.1})$$

where $\mathbf{A}_i = \mathbf{A}$, \mathbf{B} , \mathbf{C} for $i = 1, 2, 3$, respectively, and

$$\Theta = k_x u + k_y v + k_z w, \\ k_x = \frac{\partial \xi_i}{\partial x}, \quad k_y = \frac{\partial \xi_i}{\partial y}, \quad k_z = \frac{\partial \xi_i}{\partial z}, \quad \xi_i = (\xi, \eta \text{ or } \zeta) \quad \text{for } i = 1, 2, 3.$$

A similarity transform for the Jacobian matrix is introduced,

$$\mathbf{A}_i = \mathbf{T}_i \mathbf{\Lambda}_i \mathbf{T}_i^{-1}, \quad (\text{A.2})$$

with

$$\mathbf{\Lambda}_i = \frac{1}{J} \begin{bmatrix} \Theta & 0 & 0 & 0 \\ 0 & \Theta & 0 & 0 \\ 0 & 0 & \Theta + c & 0 \\ 0 & 0 & 0 & \Theta - c \end{bmatrix}, \quad (\text{A.3})$$

where c is the scaled artificial speed of sound given by

$$c = \sqrt{\Theta^2 + \beta(k_x^2 + k_y^2 + k_z^2)}. \quad (\text{A.4})$$

The matrix of the right eigenvectors is given by

$$\mathbf{T}_i = \begin{bmatrix} 0 & 0 & -\tilde{\lambda}_4 \tilde{c} & \tilde{\lambda}_3 \tilde{c} \\ \tilde{x}_2 & \tilde{x}_1 & u - \tilde{\lambda}_4 \tilde{k}_x & u - \tilde{\lambda}_3 \tilde{k}_x \\ \tilde{y}_2 & \tilde{y}_1 & v - \tilde{\lambda}_4 \tilde{k}_y & v - \tilde{\lambda}_3 \tilde{k}_y \\ \tilde{z}_2 & \tilde{z}_1 & w - \tilde{\lambda}_4 \tilde{k}_z & w - \tilde{\lambda}_3 \tilde{k}_z \end{bmatrix}, \quad (\text{A.5})$$

and its inverse is given by

$$\mathbf{T}_i^{-1} = \frac{1}{\tilde{c}^2} \begin{bmatrix} (\tilde{x}_1 a_2 + \tilde{y}_1 a_3 + \tilde{z}_1 a_1) & (\tilde{z}_1 d_2 - \tilde{y}_1 d_3) & (\tilde{x}_1 d_3 - \tilde{z}_1 d_1) & (\tilde{y}_1 d_1 - \tilde{x}_1 d_2) \\ -(\tilde{x}_2 a_2 + \tilde{y}_2 a_3 + \tilde{z}_2 a_1) & (\tilde{y}_2 d_3 - \tilde{z}_2 d_2) & (\tilde{z}_2 d_1 - \tilde{x}_2 d_3) & (\tilde{x}_2 d_2 - \tilde{y}_2 d_1) \\ 1/2 & \tilde{\lambda}_3 \tilde{k}_x / 2 & \tilde{\lambda}_3 \tilde{k}_y / 2 & \tilde{\lambda}_3 \tilde{k}_z / 2 \\ 1/2 & \tilde{\lambda}_4 \tilde{k}_x / 2 & \tilde{\lambda}_4 \tilde{k}_y / 2 & \tilde{\lambda}_4 \tilde{k}_z / 2 \end{bmatrix}, \quad (\text{A.6})$$

where

$$\tilde{x}_1 = \frac{x_{\xi_{i+1}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}}, \quad \tilde{y}_1 = \frac{y_{\xi_{i+1}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}}, \quad \tilde{z}_1 = \frac{z_{\xi_{i+1}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}},$$

$$\tilde{x}_2 = \frac{x_{\xi_{i+2}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}}, \quad \tilde{y}_2 = \frac{y_{\xi_{i+2}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}}, \quad \tilde{z}_2 = \frac{z_{\xi_{i+2}} J^{1/2}}{(k_x^2 + k_y^2 + k_z^2)^{1/4}},$$

$\xi_{i+1} = \eta, \zeta, \text{ or } \xi$, and $\xi_{i+2} = \zeta, \xi, \text{ or } \eta$, for $i = 1, 2, \text{ and } 3$, respectively,

$$\tilde{\lambda}_3 = \tilde{\Theta} + \tilde{c}, \quad \tilde{\lambda}_4 = \tilde{\Theta} - \tilde{c},$$

$$a_1 = \tilde{k}_x v - \tilde{k}_y u, \quad a_2 = \tilde{k}_y w - \tilde{k}_z v, \quad a_3 = \tilde{k}_z u - \tilde{k}_x w,$$

$$d_1 = \tilde{k}_x \beta + \tilde{\Theta} u, \quad d_2 = \tilde{k}_y \beta + \tilde{\Theta} v, \quad d_3 = \tilde{k}_z \beta + \tilde{\Theta} w,$$

$$\tilde{\Theta} = \tilde{k}_x u + \tilde{k}_y v + \tilde{k}_z w, \quad \tilde{c} = \sqrt{\tilde{\Theta}^2 + \beta},$$

$$\tilde{k}_x = \frac{k_x}{\sqrt{k_x^2 + k_y^2 + k_z^2}}, \quad \tilde{k}_y = \frac{k_y}{\sqrt{k_x^2 + k_y^2 + k_z^2}}, \quad \tilde{k}_z = \frac{k_z}{\sqrt{k_x^2 + k_y^2 + k_z^2}}.$$

ACKNOWLEDGMENTS

This work was supported by the State Key Program for Developing Basic Sciences, G1999032801, and the National Natural Science Foundation of China. Grant No. 10172089.

REFERENCES

1. F. Harlow and J. Welch, Numerical calculation of time-dependent viscous incompressible flow for fluid with free surface, *Phys. Fluids* **8**, 2182(1965).
2. A. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* **22**, 745 (1968).

3. S. Patanker, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
4. J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* **7**, 870 (1986).
5. L. Huang, Numerical solution of the unsteady incompressible Navier–Stokes equations on the curvilinear half-staggered mesh, *J. Comput. Math.* **18**, 521 (2000).
6. M. Rosenfeld and D. Kwak, Multi-grid acceleration of fractional step solvers of incompressible Navier–Stokes equations in generalized curvilinear coordinate systems, *AIAA J.* **31**(10), 1792 (1993).
7. S. Vanka, Block-implicit multigrid solution of the Navier–Stokes equations in primitive variables, *J. Comput. Phys.* **65**, 138 (1986).
8. A. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).
9. S. Rogers and D. Kwak, Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations, *AIAA J.* **28**, 253 (1990).
10. S. Rogers, D. Kwak, and C. Kiris, Steady and unsteady solutions of the incompressible Navier–Stokes equations, *AIAA J.* **29**, 603 (1991).
11. W. Briley, S. Neerarambam, and D. Whitfield, Implicit lower–upper/approximate-factorization schemes for incompressible flows, *J. Comput. Phys.* **128**, 32 (1996).
12. H. Liu and K. Kawachi, A numerical study of insect flight, *J. Comput. Phys.* **146**, 124 (1998).
13. P. Hartwich and C. Hsu, *High-Resolution Upwind Schemes for the Three-Dimensional Incompressible Navier–Stokes Equations*, AIAA Paper 87-0547 (AIAA Press, Washington DC, 1987).
14. L. Yuan and D. Xing, Comparison of the experimental and computational study of the effects of vortex control by deflecting flaps on a highly swept delta wing, *Chin. J. Aero. Astro.* **11**, 257 (1998).
15. J. Y. Yang, S. C. Yang, Y. N. Chen, and C. A. Hsu, Implicit weighted ENO schemes for three-dimensional incompressible Navier–Stokes equations, *J. Comput. Phys.* **146**, 464 (1998).
16. D. Kwak, J. Chang, S. Shanks, and S. Chakravarthy, A three-dimensional incompressible Navier–Stokes flow solver using primitive variables, *AIAA J.* **24**, 390 (1986).
17. S. Rogers, J. Chang, and D. Kwak, A diagonal algorithm for the method of pseudocompressibility, *J. Comput. Phys.* **73**, 364 (1987).
18. S. Yoon and D. Kwak, Three-dimensional incompressible Navier–Stokes solver using lower-upper symmetric-Gauss–Seidel algorithm, *AIAA J.* **29**, 874 (1991).
19. S. Rogers, Comparison of implicit schemes for the incompressible Navier–Stokes equations, *AIAA J.* **33**, 2066 (1995).
20. C. Merkle and M. Athavale, *Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility*, AIAA Paper 87-1137 (AIAA Press, Washington, DC, 1987).
21. E. Turkel, Preconditioning methods for solving the incompressible and low-speed compressible equations, *J. Comput. Phys.* **72**, 277 (1987).
22. Y. Choi and C. Merkle, The application of preconditioning in viscous flows, *J. Comput. Phys.* **105**, 207 (1993).
23. A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* **31**, 333 (1977).
24. U. Trottenberg, C. Oosterlee, and A. Chüller, *Multigrid* (Academic Press, London, 2001).
25. A. Jameson and Y. Yoon, Multigrid solution of the Euler equations using implicit schemes, *AIAA J.* **24**, 1737 (1986).
26. I. Demirdzic, Z. Lilek, and M. Peric, Fluid flow and heat transfer problem for non-orthogonal grids: Benchmark solution, *Int. J. Numer. Methods Fluids* **15**, 329 (1992).
27. C. Oosterlee and H. Ritzdorf, Flux difference splitting for the three-dimensional steady incompressible Navier–Stokes equations in curvilinear coordinates, *Int. J. Numer. Methods Fluids* **23**, 347 (1996).
28. C. Lan and M. Liang, Multigrid methods for incompressible heat flow problems with an unknown interface, *J. Comput. Phys.* **152**, 55 (1999).
29. J. Farmer, L. Martinelli, and A. Jameson, Fast multigrid method for solving incompressible hydrodynamic problems with free interfaces, *AIAA J.* **32**, 1175 (1994).

30. F. Lin and F. Sotiropoulos, Strongly-coupled multigrid method for 3D incompressible flows using near-wall turbulence closures, *J. Fluids Eng.* **119**, 314 (1997).
31. C. Sheng, L. Taylor, and D. Whitfield, Multigrid algorithm for three-dimensional incompressible high-Reynolds number turbulent flows, *AIAA J.* **33**, 2073 (1995).
32. D. Drikakis, O. Iliev, and D. Vassileva, A nonlinear multigrid method for the three-dimensional incompressible Navier–Stokes equations, *J. Comput. Phys.* **146**, 301 (1998).
33. D. Drikakis, O. Iliev, and D. Vassileva, Acceleration of multigrid flow computations through dynamic adaptation of the smoothing procedure, *J. Comput. Phys.* **165**, 566 (2000).
34. C. Liu, X. Zheng, and C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* **139**, 35 (1998).
35. S. Chakravarthy, *High Resolution Upwind Formulations for the Navier–Stokes Equations*, VKI Lecture Series on Computational Fluid Dynamics, 1988-05, Von Karman Institute for Fluid Dynamics, 1988.
36. D. Drikakis and F. Durst, Parallelization of inviscid and viscous flow solver, *Int. J. Comput. Fluid Dyn.* **3**, 101 (1994).
37. M. Schmatz, A. Brenneis, and A. Eberle, Verification of an implicit relaxation method for steady and unsteady viscous flow problems, in NATO AGARD Symposium on Validation of Computational Fluid Dynamics, Lisbon, Portugal, 1988, *AGARD-CP-437*. (1988) pp. 15-1–15-33.
38. S. Yoon and A. Jameson, Lower–upper implicit schemes with multiple grids for the Euler equations, *AIAA J.* **25**, 929 (1987).
39. P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
40. B. van Leer, Upwind-difference methods for aerodynamic problems governed by the Euler equations, *Lecture Notes Appl. Math.* **22**, 327 (1985).
41. W. Anderson, J. Thomas, and B. van Leer, *A Comparison of Finite Volume Flux Vector Splitting for the Euler Equations*, AIAA Paper 85-0122 (AIAA Press, Washington, DC, 1985).
42. T. Pulliam and D. Chaussee, A diagonal form of an implicit approximate factorization algorithm, *J. Comput. Phys.* **39**, 347 (1982).
43. K. Mohseni and T. Colonius, Numerical treatment of polar coordinates singularities, *J. Comput. Phys.* **157**, 787 (2000).
44. J. Humphrey, A. Taylor, and J. Whitelaw, Laminar flow in a square duct of strong curvature, *J. Fluid Mech.* **83**, Part 3, 509 (1977).
45. F. White, *Viscous Fluid Flow* (McGraw–Hill, New York, 1974), p. 123.
46. K. Wang, H. Zhou, C. Hu, and S. Harrington, Three-dimensional separated flow structure over prolate spheroids, *Proc. R. Soc. London A* **421**, 73 (1990).
47. W. Su, B. Tao, and L. Xu, Experimental investigation of three-dimensional separated flow over a prolate spheroid, in *Proceedings of the 1st International Conference on Experimental Fluid Mechanics, Chengdu, China, 1991*, edited by F. G. Zhuang (International Academic Publishers, Beijing, 1992), pp. 205–210.
48. H. Kreplin, H. Vollmers, and H. Meier, Measurements of the wall shear stress on an inclined prolate spheroid, *Z. Flugwiss. Weltraumforsch.* **6**, 248 (1982).
49. A. Panaras and J. Steger, A thin-layer solution of the flow about a prolate spheroid, *Z. Flugwiss. Weltraumforsch.* **12**, 173 (1988).
50. V. Vatsa, J. Thomas, and B. Wedan, Navier–Stokes computations of a prolate spheroid at angle of attack, *AIAA J.* **26**, 986 (1989).
51. N. Pierce and J. Alonso, Efficient computation of unsteady viscous flows by an implicit preconditioned multigrid method, *AIAA J.* **36**, 401 (1998).