JOURNAL OF COMPUTATIONAL PHYSICS 114, 18–33 (1994)

A Non-staggered Grid, Fractional Step Method for Time-Dependent Incompressible Navier–Stokes Equations in Curvilinear Coordinates

YAN ZANG, ROBERT L. STREET, AND JEFFREY R. KOSEFF

Environmental Fluid Mechanics Laboratory, Stanford University, California 94305-4020

Received March 5, 1992; revised February 7, 1994

A numerical method for solving three-dimensional, time-dependent incompressible Navier–Stokes equations in curvilinear coordinates is presented. The non-staggered-grid method originally developed by C. M. Rhie and W. L. Chow (*AIAA J.* **21**, 1525 (1983)) for steady state problems is extended to compute unsteady flows. In the computational space, the Cartesian velocity components and the pressure are defined at the center of a control volume, while the volume fluxes are defined at the mid-point on their corresponding cell faces. The momentum equations are integrated semi-implicitly by the approximate factorization technique. The intermediate velocities are interpolated onto the faces of the control volume to form the source terms of the pressure Poisson equation, which is solved iteratively with a multigrid method. The compatibility condition of the pressure Poisson equation is satisfied in the same manner as in a staggered-grid method; mass conservation can be satisfied to machine accuracy. The pressure boundary condition is derived from the momentum equations. Solutions of both steady and unsteady problems including the large eddy simulation of a rotating and stratified upwelling flow in an irregular container established the favorable accuracy and efficiency of the present method.

1. INTRODUCTION

Solution methods for the incompressible Navier–Stokes equations have been successfully developed in Cartesian coordinate systems in the past (e.g., [1, 2]). In recent years, more effort has been put into developing solution methods in curvilinear coordinates. Rogers *et al.* [3] solved both steady-state and time-dependent problems using an artificial compressibility method originally proposed by Chorin [4]. The advantage of introducing an artificial compressibility into the continuity equation is that efficient solution algorithms developed for compressible flows can be utilized to compute incompressible flows. A disadvantage of the method is that the system of equations may become highly stiff in a time-dependent situation [5] for which an implicit method may have to be employed. The use of an implicit method in Rogers *et al.* requires a large amount of memory usage (180 variables stored per grid point [3]), which poses a severe limitation in three-dimensional computations. The artificial compressibility method, following solution techniques in compressible flows, naturally adopts the Cartesian velocities as the primary dependent variables, together with the traditional non-staggered grid in which all the flow variables are defined at the cell center.

Another popular method is the fractional step or the projection method [1, 6, 7]. This method splits the numerical operators and achieves pressure–velocity coupling through solving a Poisson-like elliptic equation for pressure (i.e., the pressure Poisson equation). Unlike the artificial compressibility method, the fractional step method may be coupled with different choices of dependent variables and grid layouts. For example, Cartesian, contravariant, or covariant velocity components may be employed as the dependent variables. In addition, either a staggered grid [8] or a non-staggered grid may be used. A sketch of the definition of various grid layouts in two dimensions is shown in Fig. 1.

Cartesian velocity components coupled with a staggered grid have been employed by several researchers [9–11] (Fig. 1a). In three-dimensional problems, all three Cartesian velocity components are defined on each control volume face resulting in the definition of nine velocities and the solution of nine momentum equations per control volume. Moreover, a special interpolation scheme needs to be used to prevent the decoupling of the covariant velocity components from the pressure field [9]. This method was employed by the present authors in [11] but was abandoned later due to its inefficiency. Partly to avoid the above problems, a different staggered-grid setup was developed [12, 13] which defines only one Cartesian velocity component on each control-volume face (Fig. 1b). This method only requires the solution of three momentum equations per control volume; however, its accuracy is not independent of the grid orientation [14]. In Fig. 1b, we see that when the grid turns by 90°, the Cartesian velocity defined on a cell face becomes parallel to the face. As a result, the volume fluxes through the faces of the cell have to

0021-9991/94 $6.00

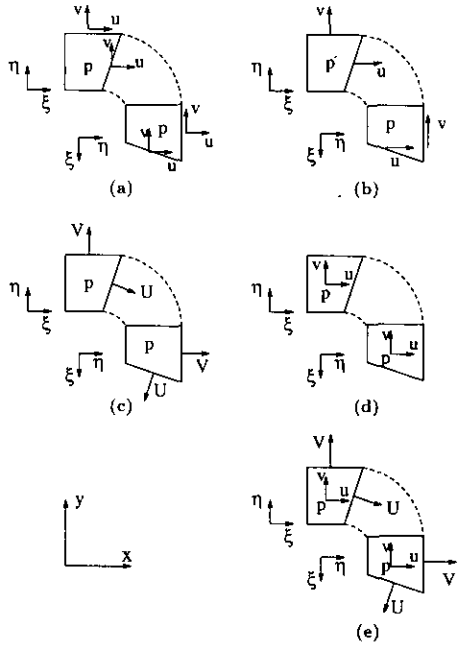


FIG. 1. Effects of cell orientation on variables in various grid layouts in curvilinear coordinates: (a) staggered grid with all the Cartesian velocity components defined on each cell face; (b) staggered grid with only one Cartesian velocity component defined on each cell face; (c) staggered grid with the volume flux defined on its corresponding cell face; (d) traditional non-staggered grid; (e) new non-staggered grid. Notations are x, y: Cartesian coordinates; ξ, η: curvilinear coordinates; u, v: Cartesian velocity components; U, V: volume fluxes; p: pressure.

be interpolated, an undesirable fact which tends to diminish the advantage of a staggered grid. To avoid a negative impact on the solution accuracy, special care needs to be taken [13]. Rosenfeld *et al.* [15] employed the volume fluxes (the contravariant velocity multiplied by the cell volume) defined on cell faces as primary variables and discretized the governing equations on a staggered grid using the finite volume method (Fig. 1c). This approach mimics the standard staggered-grid method in Cartesian coordinates in which the compatibility condition of the pressure Poisson equation is satisfied automatically. The discretization, however, is quite complicated and the effort to solve three momentum equations for the volume fluxes is equivalent to solving nine momentum equations for the Cartesian velocity components. Moreover, in order to avoid storing too many geometric quantities which arise from the formulation, approximations have to be made to find these quantities at different locations in a control volume. In another development, Karki and Patankar [16] presented a method using a staggered grid and the covariant velocity components as primary variables.

The use of Cartesian velocity components as the primary dependent variables has the advantage that the governing equations remain in a relatively simple form and can be put into strong-conservation-law form. On the other hand, the use of covariant or contravariant velocity components (or volume fluxes) results in extra source terms and more complicated equations [13].

In a staggered grid, the cells that enclose the pressure node and the velocity nodes are different. As a result, in three-dimensional curvilinear coordinates, a staggered-grid method may require storage of as many as seven sets of metrics per cell, i.e., metrics defined at the center, the mid-point of the three cell faces, and the mid-point of the three edges [10, 12], although a more effective arrangement in a control volume approach may reduce this number to about half as many. In transforming the Navier–Stokes equation into a non-orthogonal coordinate system, we usually have a set of 16 metrics in a finite volume formulation which includes nine components of the three vector surface areas, one volume or the inverse of the Jacobian, and six nontrivial elements of the symmetric mesh skewness tensor. The result is that a staggered-grid method in curvilinear coordinates may require a large amount of computer memory to store the metrics. Simple averaging of metrics defined at cell centers onto cell faces and edges is generally considered to result in a loss of accuracy. This has been experienced by the present authors. Although Rosenfeld *et al.* [15] indicated that their averaging of geometric quantities satisfies the geometric conservation law, approximations had to be used in the evaluation of certain geometric terms which may have negative effects on accuracy.

In order to avoid the above difficulties, researchers in recent years have investigated methods based on non-staggered grids [17–23]. Consistent differencing schemes for the pressure Poisson equation and its boundary conditions have been derived for both Cartesian and curvilinear coordinates to satisfy the compatibility condition on a non-staggered grid [17, 18]. Another approach, originally proposed by Rhie and Chow [19] and subsequently modified by Perić *et al.* [20] and Majumdar [21], is to employ special interpolations to obtain a compact stencil for the pressure gradient on a non-staggered grid to avoid pressure–velocity decoupling. Only three instead of nine momentum equations need to be solved. Most studies using this approach have coupled it with the SIMPLE algorithm [2] and solved steady-state problems. This so-called "colocated-grid method" [20] or "momentum interpolation method" [21] has demonstrated good stability and accuracy in solving steady-state problems.

Solution methods for the incompressible Navier–Stokes equations based on a non-staggered grid have long had the reputation of producing spurious oscillations in the pressure field, i.e., the "checkerboard" pattern [2]. One of the fundamental causes is that, in a traditional non-staggered grid (Fig. 1d), all the variables are defined at the cell center, and as a consequence, a straightforward discretization of the continuity equation does not enforce mass conservation in the cell and causes decoupling of the pressure field. The

non-staggered-grid method of Rhie and Chow prevents the decoupling by defining the volume flux on its corresponding face of the cell in addition to the Cartesian velocity components at the cell center (Fig. 1e). In this way, both the momentum and the continuity equations are enforced in the same control volume. Analysis has been carried out to compare the elliptic property of several non-staggered-grid methods [23]. Results show that Rhie and Chow's method is strongly elliptic in the sense that all realizable modes in the wavenumber space on a given grid can be uniquely determined. This uniqueness prevents energy from accumulating at the grid scale and, thus, guarantees that solutions are free from spurious pressure oscillations. Because of the non-staggered-grid layout, the accuracy of this method is not affected by grid orientations, and so it is directly applicable to curved domains with, for example, 90° turns.

Based on insights from the above, we have extended the non-staggered-grid method which was developed for steady flows to solve the time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. One major purpose is to employ recognized techniques such as the fractional step procedure, approximate factorization, QUICK, high-order interpolation, and multigrid to build an overall efficient and accurate solution method. A straightforward formulation is presented which naturally couples the fractional step method with the non-staggered grid without special treatment. The compatibility condition for the pressure Poisson equation is satisfied automatically, which makes it possible to enforce mass conservation to machine accuracy.

Most published results which employ the Rhie and Chow non-staggered-grid involve steady flows at low Reynolds numbers (Re $\sim$ 10–1000). Another purpose of the present work, therefore, is to provide a detailed examination and to validate the accuracy and efficiency of this method as it is applied to both steady and unsteady flows. This is achieved by comparing the present results for several benchmark tests to those obtained using well-established staggered-grid methods.

Finally, we present results of the large eddy simulation of a turbulent rotating and stratified flow in an irregular container which resembles the coastal upwelling phenomenon. This case represents a challenge in testing the capability of the present method in treating unsteady and both physically and geometrically complex flows.

The governing equations and the coordinate transformation are presented in Section 2. The numerical method is described in Section 3. In Section 4, results from five test cases are presented. The test cases include one with an exact solution, two in Cartesian coordinates, and two in curvilinear coordinates. The large eddy simulation results of the upwelling flow is presented in Section 5. A summary is given in Section 6.

2. GOVERNING EQUATIONS

We present the governing Navier–Stokes and continuity equations in the constant viscosity and constant density form,

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j u_i) = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2}$$

where $i, j = 1, 2, 3$; u_i represents the Cartesian velocity components; p is the pressure divided by fluid density; ν is the kinematic viscosity. The present numerical method is directly applicable to incompressible, variable density and/or rotating fluid (see Section 5 and Ref. [24]). Equations (1) and (2) are transformed into curvilinear coordinates in strong-conservation-law form as

$$\frac{\partial U_m}{\partial \xi_m} = 0 \tag{3}$$

$$\frac{\partial (J^{-1} u_i)}{\partial t} + \frac{\partial F_{im}}{\partial \xi_m} = 0, \tag{4}$$

where the flux is

$$F_{im} = U_m u_i + J^{-1} \frac{\partial \xi_m}{\partial x_i} p - \nu G^{mn} \frac{\partial u_i}{\partial \xi_n}; \tag{5}$$

J^{-1} is the inverse of the Jacobian or the volume of the cell; U_m is the volume flux (contravariant velocity multiplied by J^{-1}) normal to the surface of constant ξ_m; and G^{mn} is called the "mesh skewness tensor." These quantities are

$$U_m = J^{-1} \frac{\partial \xi_m}{\partial x_j} u_j \tag{6}$$

$$J^{-1} = \det\left(\frac{\partial x_i}{\partial \xi_j}\right) \tag{7}$$

$$G^{mn} = J^{-1} \frac{\partial \xi_m}{\partial x_j} \frac{\partial \xi_n}{\partial x_j}. \tag{8}$$

3. NUMERICAL METHOD

3.1. Discretization

We employ the non-staggered-grid layout shown in Fig. 1e. The pressure and the Cartesian velocity components are defined at the center and the volume fluxes are defined at the mid-point of their corresponding faces of the control volume in the computational space (Fig. 2). Following Kim

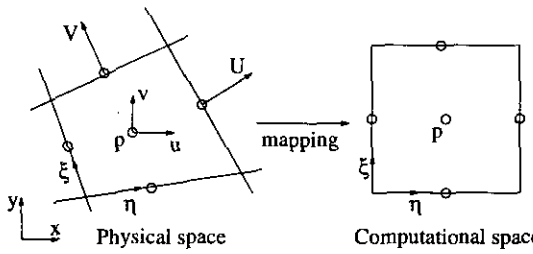


FIG. 2. A control volume of the non-staggered grid and the mapping in two dimensions.

and Moin [1], we use a semi-implicit time-advancement scheme with the Adams–Bashforth method for the explicit terms and the Crank–Nicolson for the implicit terms. The discretized equations are

$$\frac{\delta U_m}{\delta \xi_m} = 0 \tag{9}$$

$$J^{-1}\frac{u_i^{n+1}-u_i^n}{\Delta t} = \frac{3}{2}(C_i^n + D_E(u_i^n)) - \frac{1}{2}(C_i^{n-1} + D_E(u_i^{n-1})) + R_i(p^{n+1}) + \frac{1}{2}(D_I(u_i^{n+1} + u_i^n)), \tag{10}$$

where $\delta/\delta\xi_m$ represents discrete finite difference operators in the computational space; superscripts represent the time step; C_i represents the convective terms; R_i is the discrete operator for the pressure gradient terms; and D_E and D_I are discrete operators representing, respectively, the explicitly treated off-diagonal viscous terms and the implicitly treated diagonal viscous terms. They are

$$C_i = -\frac{\delta}{\delta \xi_m}(U_m u_i) \tag{11}$$

$$R_i = -\frac{\delta}{\delta \xi_m}\left(J^{-1}\frac{\delta \xi_m}{\delta x_i}\right) \tag{12}$$

$$D_I = \frac{\delta}{\delta \xi_m}\left(\nu G^{mn}\frac{\delta}{\delta \xi_n}\right), \qquad m = n \tag{13}$$

$$D_E = \frac{\delta}{\delta \xi_m}\left(\nu G^{mn}\frac{\delta}{\delta \xi_n}\right), \qquad m \neq n. \tag{14}$$

The diagonal viscous terms are treated implicitly in order to remove the viscous stability limit. It is well known that in the computation of high Reynolds number flows, very fine grids need to be used near a solid boundary. Moreover, in turbulent calculations, the eddy viscosity may be several orders of magnitude larger than the molecular viscosity. This makes the implicit treatment of the viscous terms imperative because the viscous stability limit is much more restrictive than the inviscid CFL (Courant–Friedrichs–Lewy) condition near solid boundaries. The off-diagonal viscous terms usually have less significant contribution, unless the grid is severely skewed. They are treated explicitly here in order to yield a simpler structure of the left-hand-side (LHS) matrix of the momentum equation. In highly skewed or highly irregular grids, this explicit treatment of the off-diagonal viscous terms may reduce the maximum allowable time step. Nevertheless, highly skewed or irregular grids should be avoided whenever possible in order to preserve the accuracy of the numerical method. Sometimes it is difficult to generate a smooth grid in complex geometries, especially in three dimensions. This problem, however, may be overcome by using domain decomposition and composite grids [11].

Except for the convective terms, all the spatial derivatives are approximated with second-order central differences. The convective terms are discretized using a variation of QUICK [25, 26] which calculates the face value from the nodal values using a quadratic upwind interpolation. Several researchers have compared the accuracy of QUICK to that of central differencing, first-order upwind, the hybrid scheme, and the "skew upwind differencing scheme" of Raithby [27], and found that QUICK produces superior results [28–32]. The "power-law" scheme [2] is accurate under normal conditions but reduces to first-order accuracy when convection is dominant. Although QUICK does not rule out the appearance of over- and undershoots, no numerical oscillations have been observed in any of the homogeneous flow cases presented here. In the simulation of the stratified upwelling flow in Section 5, the convective term in the scalar transport equation is discretized using SHARP [33]. The monotonic feature of SHARP is desirable in this case because a sharp density gradient exists in the flow domain. Because SHARP is computationally more expensive than QUICK, it is only used in solving the scalar transport equation. The upwinding of QUICK is carried out by computing positive and negative volume fluxes, $(U_m + |U_m|)/2$ and $(U_m - |U_m|)/2$, and using the generic stencil.

3.2. Fractional Step Method

Application of the fractional step method to (10) leads to the following predictor–corrector solution procedure

1. Predictor:

$$\left(I - \frac{\Delta t}{2J^{-1}}D_I\right)(u_i^* - u_i^n) = \frac{\Delta t}{J^{-1}}\left[\frac{3}{2}(C_i^n + D_E(u_i^n)) - \frac{1}{2}(C_i^{n-1} + D_E(u_i^{n-1})) + D_I(u_i^n)\right]; \tag{15}$$

2. Corrector:

$$u_i^{n+1} - u_i^* = \frac{\Delta t}{J^{-1}} [R_i(\phi^{n+1})], \tag{16}$$

where I is the identity matrix. The variable u_i^* is called the "intermediate velocity" which is not constrained by continuity. The variable ϕ is related to p by

$$R_i(p) = \left(J^{-1} - \frac{\Delta t}{2} D_I\right)\left(\frac{R_i(\phi)}{J^{-1}}\right). \tag{17}$$

To invert the matrix on the LHS of (15), we employ the approximate factorization technique [34, 35] in which (15) is rewritten as

$$\left(I - \frac{\Delta t}{2J^{-1}}(D_1 + D_2 + D_3)\right)(u_i^* - u_i^n) = \frac{\Delta t}{J^{-1}}\left[\frac{3}{2}(C_i^n + D_E(u_i^n)) - \frac{1}{2}(C_i^{n-1} + D_E(u_i^{n-1})) + D_I(u_i^n)\right] \tag{18}$$

where

$$D_k = \frac{\delta}{\delta \xi_k}\left(\nu G^{kk} \frac{\delta}{\delta \xi_k}\right) \tag{19}$$

and $k = 1, 2, 3$. No summation on k is taken in the above equation. The LHS of (18) is then factorized as

$$\left(I - \frac{\Delta t}{2J^{-1}} D_1\right)\left(I - \frac{\Delta t}{2J^{-1}} D_2\right)\left(I - \frac{\Delta t}{2J^{-1}} D_3\right)(u_i^* - u_i^n) = \frac{\Delta t}{J^{-1}}\left[\frac{3}{2}(C_i^n + D_E(u_i^n)) - \frac{1}{2}(C_i^{n-1} + D_E(u_i^{n-1})) + D_I(u_i^n)\right]. \tag{20}$$

It is easy to show that the error of the above factorization is of order $O(\Delta t^3)$. The inversion of the LHS of (20) requires solving three tridiagonal matrices. The boundary condition for the intermediate velocity u_i^* derived in [1] is extended to curvilinear coordinates which requires that

$$u_i^* = u_i^n - \frac{\Delta t}{J^{-1}} R_i(\phi^n) + O(\Delta t^2). \tag{21}$$

3.3. Pressure Poisson Equation

The variable ϕ^{n+1} is obtained by solving the pressure Poisson equation which is derived discretely by the following procedure. First we derive the equation for the volume flux U_m^{n+1}. If we imagine that we apply the corrector step of the fractional step method (16) to the Cartesian velocity components defined on a certain face of the control volume, we have

$$(u_i^{n+1})_{\text{face}} = (u_i^*)_{\text{face}} - \Delta t \left(\frac{\delta \xi_m}{\delta x_i}\frac{\delta \phi^{n+1}}{\delta \xi_m}\right)_{\text{face}}. \tag{22}$$

The above equation is different from (16) in that, instead of being written in the strong-conservation-law form, the pressure gradient is written in the chain-rule-conservation-law form. Combining (22) with (6), we obtain the equations for U_m^{n+1} as

$$U_m^{n+1} = U_m^* - \Delta t \left(G^{mn}\frac{\delta \phi^{n+1}}{\delta \xi_n}\right), \tag{23}$$

where $U_m^* = J^{-1}(\delta \xi_m / \delta x_j) u_j^*$ is called the intermediate volume flux. Since the intermediate velocity u_j^* is defined at the cell center, while the fluxes U_m^* and U_m^{n+1} are defined on the cell faces, u_j^* has to be interpolated onto the cell faces in order to compute U_m^*. In the present work, we use a third-order accurate upwind quadratic interpolation similar to the QUICK formulation [25] to obtain u_j^* on the cell faces.

By substituting (23) into (9), we obtain the pressure Poisson equation for ϕ^{n+1} as

$$\frac{\delta}{\delta \xi_m}\left(G^{mn}\frac{\delta \phi^{n+1}}{\delta \xi_n}\right) = \frac{1}{\Delta t}\frac{\delta U_m^*}{\delta \xi_m}. \tag{24}$$

The above derivation results in a pressure Poisson equation whose coefficients consist only of the mesh skewness tensor G^{mn}. This elliptic equation is solved iteratively using a multigrid method [26, 36]. The four-color ZEBRA sweep proposed in [15] combined with the line-by-line Gauss–Seidel method is used as the smoother. The four-color ZEBRA sweep makes the pressure solver completely vectorizable in three dimensions and preserves the convergence property of a line-by-line Gauss–Seidel sweep. The present multigrid method employs V cycles and performs two or three smoothing iterations on each level. The fast convergence rate of the multigrid method makes it possible to convergence the pressure Poisson equation to machine zero and, thus, to satisfy mass conservation exactly. The efficiency of the present multigrid method is demonstrated in Section 4.

In Cartesian or orthogonal coordinates, no pressure boundary condition is required if a staggered or the present non-staggered grid is used. In a non-orthogonal coordinate system, however, a pressure boundary condition is required in both types of grid layout. This is shown by examining the derivation of the pressure Poisson equation for a control

volume adjacent to a grid boundary. Figure 3 is a sketch of the grid near a boundary which is at $i=\frac{3}{2}$. For convenience, we consider the two-dimensional case in the following illustration. Writing Eq. (23) in two dimensions for the volume fluxes defined on the four faces of the cell $(2, j)$ adjacent to a grid boundary and dropping the superscript $n+1$, we have

$$U_{3/2,j} = U^*_{3/2,j} - \Delta t[G^{11}_{3/2,j}(\phi_{2,j}-\phi_{1,j}) + \tfrac{1}{4}G^{12}_{3/2,j}(\phi_{2,j+1}+\phi_{1,j+1}-\phi_{2,j-1}-\phi_{1,j-1})] \tag{25}$$

$$U_{5/2,j} = U^*_{5/2,j} - \Delta t[G^{11}_{5/2,j}(\phi_{3,j}-\phi_{2,j}) + \tfrac{1}{4}G^{12}_{5/2,j}(\phi_{3,j+1}+\phi_{2,j+1}-\phi_{3,j-1}-\phi_{2,j-1})] \tag{26}$$

$$V_{2,j-1/2} = V^*_{2,j-1/2} - \Delta t[G^{22}_{2,j-1/2}(\phi_{2,j}-\phi_{2,j-1}) + \tfrac{1}{4}G^{12}_{2,j-1/2}(\phi_{3,j}+\phi_{3,j-1}-\phi_{1,j}-\phi_{1,j-1})] \tag{27}$$

$$V_{2,j+1/2} = V^*_{2,j+1/2} - \Delta t[G^{22}_{2,j+1/2}(\phi_{2,j+1}-\phi_{2,j}) + \tfrac{1}{4}G^{12}_{2,j+1/2}(\phi_{3,j+1}+\phi_{3,j}-\phi_{1,j+1}-\phi_{1,j})]. \tag{28}$$

In the above equations, values of ϕ at the fictitious points $(i=1)$ outside the boundary appear in the equations for $U_{3/2,j}$, $V_{2,j-1/2}$, and $V_{2,j+1/2}$. When the pressure Poisson equation is derived for control volume $(2, j)$, Eqs. (26), (27), and (28) are substituted into Eq. (9) for the three interior fluxes $U_{5/2,j}$, $V_{2,j-1/2}$, and $V_{2,j+1/2}$. For the boundary flux $U_{3/2,j}$, Eq. (25) is not used; instead, the physical boundary condition for velocity is imposed. It is easy to show that since the physical boundary condition is imposed for the volume flux on the boundary when the pressure Poisson equation is derived, global mass conservation and the compatibility condition are automatically satisfied [24].

To obtain the pressure value at the fictitious points, we employ the momentum equation normal to the boundary. By applying Eq. (23) on the cell face $i=\frac{3}{2}$ in Fig. 3, we have

$$G^{11}(\phi^{n+1}_{2,j}-\phi^{n+1}_{1,j}) + G^{12}\frac{\delta\phi^{n+1}}{\delta\xi_2} + G^{13}\frac{\delta\phi^{n+1}}{\delta\xi_3} = \frac{1}{\Delta t}(U^*_{3/2,j}-U^{n+1}_{3/2,j}), \tag{29}$$

where $U^{n+1}_{3/2,j}$ is the imposed boundary volume flux. The value of $U^*_{3/2,j}$ is computed from $u^*_{3/2,j}$ which is extrapolated from the interior u_i^* values with a third-order accurate scheme. The use of extrapolation to compute U^*_ξ at the boundary is consistent with the use of interpolation to obtain U^*_ξ in the interior cells. Previous authors have used simple linear extrapolation of the pressure value onto the boundary [20, 21]. Numerical experiments by the present authors of lid-driven flows in a square cavity have shown that simple pressure extrapolation produce erroneous behavior at the separation and reattachment points.

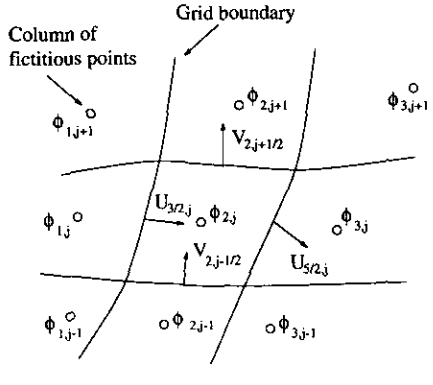


FIG. 3. Sketch of the grid near a boundary.

3.4. Overall Solution Procedure

The overall solution procedure to numerically solve the governing equations is as follows:

1. Solve for the intermediate velocity u_i^* at cell centers using (20);
2. Interpolate u_i^* onto the cell faces and compute the RHS of (24);
3. Solve (24) by the multigrid method to obtain a converged ϕ^{n+1};
4. Obtain u_i^{n+1} from (16) and U_m^{n+1} from (23) to finish one time step.

Since the viscous stability limit is removed by advancing the viscous terms implicitly, the stability of the overall numerical method is restricted by the CFL condition. The local CFL number is defined as

$$\mathrm{CFL} = \left(\frac{|u_1|}{\Delta x}+\frac{|u_2|}{\Delta y}+\frac{|u_3|}{\Delta z}\right)\Delta t = (|U_1|+|U_2|+|U_3|)\frac{\Delta t}{J^{-1}}, \tag{30}$$

where Δx, Δy, Δz are the grid spacing of the three Cartesian coordinates. The stability condition of the present method requires that

$$\max\{\mathrm{CFL}\} < \bar{C} \sim 1, \tag{31}$$

where "max{CFL}" is the maximum value obtained from (30) in the computational domain. Since the Adams–Bashforth method is absolutely unstable in the

linear convective limit, the value of $\bar{C}$ is a function of the Reynolds number. Present numerical experiments have shown that $\bar{C}=1$ yields stable solutions. However, as discussed previously, $\bar{C}$ may become smaller for highly skewed grids.

In the present code, only one set of metrics is stored per control volume. On the cell face which is normal to the coordinate line ξ_m, the vector surface area, $J^{-1}\,\delta\xi_m/\delta x_i$ $(i=1, 2, 3)$, and the three corresponding elements of the mesh skewness tensor, G^{mn} $(n=1, 2, 3)$ are stored. The cell volume or the inverse of the Jacobian, J^{-1}, is stored at the cell center. This results in a total storage of 19 metrics per control volume. No averaging of metrics is required. Quantitative comparisons of memory and CPU usage between the present non-staggered-grid code and two previously used staggered-grid codes will be given in Section 4.

The computer code is written in three dimensions. Two-dimensional and axisymmetric problems are solved by using one control volume and specifying appropriate periodic boundary conditions in the third dimension.

4. METHOD VALIDATION

4.1. Decaying Vortices

It is important to test the accuracy of a numerical method and the computer code by computing a flow with an analytical solution. The errors of the discretization can be explicitly calculated and the order of accuracy of the overall method can be clearly shown. The two-dimensional unsteady flow of decaying vortices has the following exact solution:

$$u(x, y, t) = -\cos(x)\sin(y)\,e^{-2t} \tag{32}$$

$$v(x, y, t) = \sin(x)\cos(y)\,e^{-2t} \tag{33}$$

$$p(x, y, t) = -\tfrac{1}{4}[\cos(2x)+\cos(2y)]\,e^{-4t}. \tag{34}$$

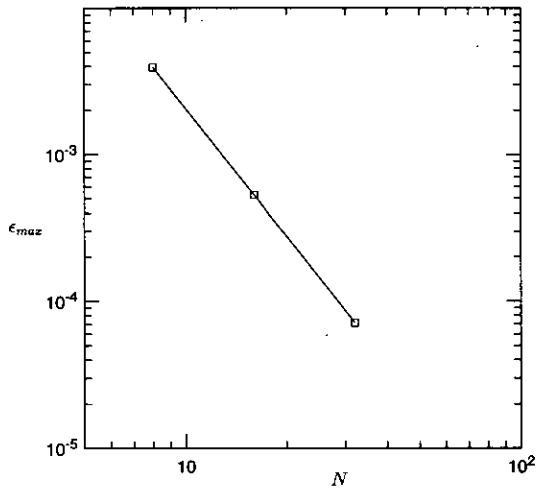


FIG. 4. Maximum relative error of u as a function of mesh refinement for the decaying vortices flow. N is the number of grid points in each dimension.

Computations are carried out in the domain $0 \leqslant x, y \leqslant \pi$. This flow has been chosen by previous researchers (e.g., [1, 3]) to test the accuracy of their numerical methods and boundary conditions.

The maximum relative error in u, $\varepsilon_{\max}$, at the dimensionless time of 1.0 was plotted in Fig. 4 as a function of mesh refinement. Similar results were obtained for v. A uniform grid was used in the computation. The time step was refined proportional to the grid spacing. On the boundaries, the exact solution was imposed. Figure 4 shows that the present method and the boundary conditions are indeed second-order accurate. The accurate prediction of this unsteady flow demonstrates the time-accurate capability of the code.

4.2. Lid-Driven Flow in a Square Cavity

Lid-driven flows in a two-dimensional square cavity at the Reynolds number of 3200 and in three-dimensional square cavities at Reynolds numbers of 3200, 7500, and 10000 (based on the lid velocity and the length of the cavity), were simulated. These Reynolds numbers were chosen because well-documented experimental as well as numerical data were available. Another reason was that in the three-dimensional case, the Reynolds numbers were sufficiently high that the flows were fundamentally unsteady, making time-accurate calculations necessary.

4.2.1. *Lid-Driven Flow in a Two-Dimensional Square Cavity*

The geometry and the boundary conditions of the lid-driven square cavity are shown in Fig. 5. Figure 6 gives the center-line velocities ($u(y)$ and $v(x)$ along the vertical and the horizontal center-line, respectively) at steady state of the driven flow in a two-dimensional cavity. A non-uniform 42×42 grid is used in the present computation. Previous results from Ghia *et al.* [37] (129×129 uniform grid) and from Perng [10] (42×42 non-uniform grid) are presented for comparison. Both 42×42 non-uniform grids yield

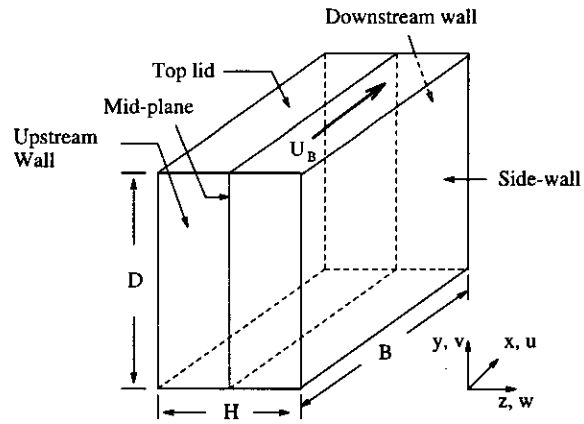


FIG. 5. Geometry and boundary conditions for the lid-driven flow in a square cavity.

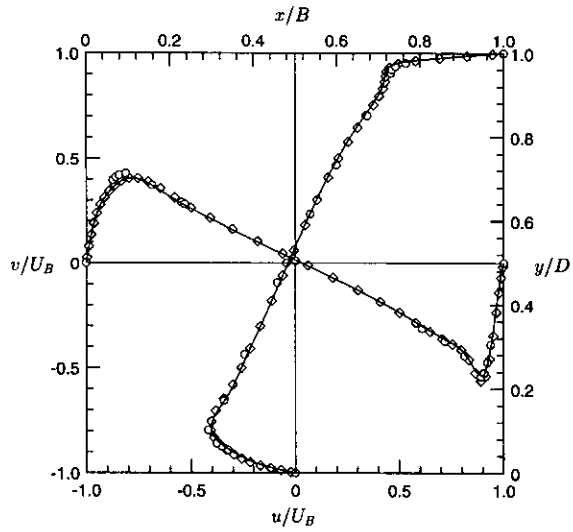


FIG. 6. Comparison of the center-line velocity profiles $u(y)$ and $v(x)$ (Re = 3200): ○, Ghia *et al.* [37]; ◇, Perng [10]; —, present.

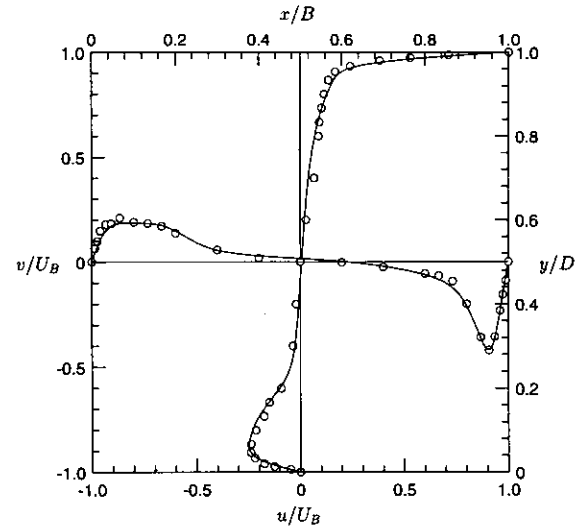


FIG. 7. Time-averaged center-line velocity profiles $u(y)$ and $v(x)$ on the mid-plane of the cubic cavity at the fully-developed state (Re = 3200): ○, Prasad *et al.* [38]; —, present.

slightly weaker flow compared with Ghia *et al.*'s fine grid calculation. The averaged difference between the present and Perng's profiles divided by the lid velocity is less than 0.5%. When the steady-state velocity field was compared with that from the standard staggered-grid computation in [10], good agreement was obtained in the size of the primary as well as the corner vortices (see Ref. [24]). The present solution accurately predicted the location of the separation and reattachment points. When the boundary pressure was determined by (29), no unphysical behavior was observed near the separation and the reattachment points. On the other hand, schemes which computed the boundary pressure by using a direct extrapolation (linear or quadratic) from the interior values resulted in erroneous behavior.

4.2.2. *Lid-Driven Flow in a Three-Dimensional Cubic Cavity*

Lid-driven flows in a three-dimensional cubic cavity at high Reynolds numbers have been investigated both experimentally and numerically [38–40]. The highly unsteady flows exhibit complex flow structures such as the Taylor–Göertler-like (TGL) vortices first observed by Koseff *et al.* [39]. This flow serves as an excellent test case for time-accurate numerical methods. We computed the starting flow in a half cubic cavity (from the endwall to the midplane) at a Reynolds number of 3200. Numerical experiments by Perng [10] have shown that at this Reynolds number the flow is essentially symmetric over the midplane of the cavity. We compared our solutions with the results in [38] for flowfields at 10, 30, 60, and 90 s after start-up. Good agreement was found (not shown here to save space). In Fig. 7, the computed center-line velocity profiles at the midplane of the cavity are compared with the experimental results [38]. The measured data are 5-min averages at the fully developed state, while the computed data are 3-min averages from 10 to 13 min after start-up. The computational grid has $34 \times 34 \times 18$ grid points, being non-uniform in the $x-y$ plane and uniform in the z direction. The averaged difference between the two sets of data divided by the lid velocity is less than 0.5%. In Fig. 8, instantaneous streamlines in a $y-z$ plane at $0.25B$ from the downstream wall show the existence of two pairs of TGL vortices in the whole cavity. In this case, 34 uniformly dis-

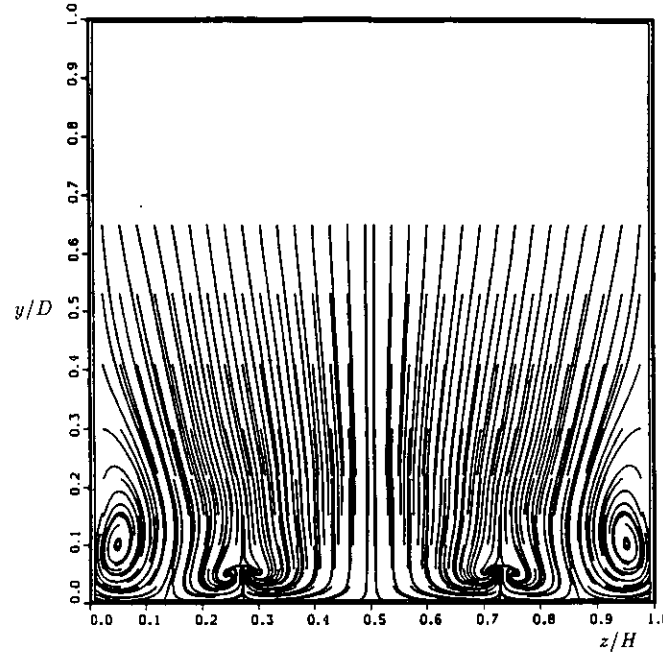


FIG. 8. Instantaneous streamlines in the $y-z$ plane $0.25B$ from the downstream wall of the cubic cavity flow, $t = 540$ s (Re = 3200).

tributed grid points are used in the z direction to obtain a better resolution of TGL vortices. Only the flow in the left half of the cubic cavity is simulated and the data is flipped onto the right half. Prasad *et al.* [38] also observed two pairs of TGL vortices in the cubic cavity near the downstream wall at this Reynolds number. The presently computed spanwise wavelength of the TGL vortices is $0.37B$ which agrees precisely with the value measured by Prasad [40], which is also $0.37B$. These vortices are meandering in the spanwise direction and are highly unsteady. They are responsible for a large portion of momentum transfer in the cavity.

Lid-driven flows in a three-dimensional cavity at Reynolds numbers of 7500 and 10000 were also computed using the present numerical method. These flows are in the transitional and locally turbulent regimes. Large eddy simulations were performed with small-scale motions being modeled by either the "dynamic eddy viscosity model" [41, 42] or the "dynamic mixed model" [43]. Both the mean and the fluctuating quantities were computed and compared with the measurements of Prasad and Koseff [44]. Excellent agreement was achieved. For details of the simulation results and the coupling of the numerical method with the subgrid-scale turbulence models; the readers are referred to [43].

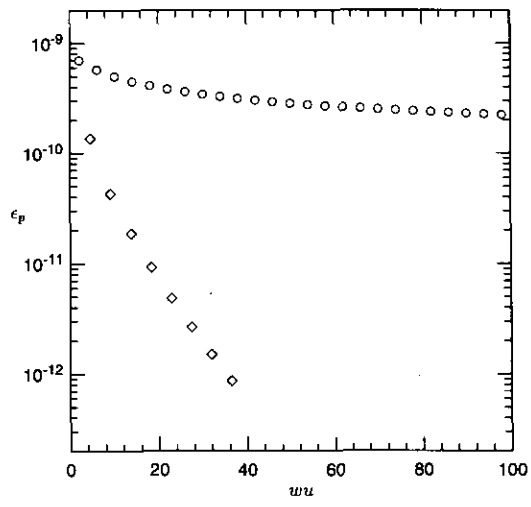


FIG. 9. Residual of the pressure equation for cubic cavity flow at the first time step: ○, point SOR; ◇, multigrid.

4.2.3. *Efficiency of the Method*

The performance of the present numerical method in terms of efficiency is now evaluated using the three-dimensional lid-driven cavity flow at the Reynolds number of 3200 which has been described in Section 4.2.2. The iterative solution of the pressure Poisson equation is the most time-consuming part in the overall solution procedure. For high Reynolds number flows and highly stretched or skewed grids, it is very difficult to converge the pressure Poisson equation to machine accuracy. Therefore, the use of a rapidly convergent scheme such as the multigrid method is a very desirable feature. Figure 9 shows the residual of the pressure Poisson equation (ε_p) as a function of the working unit (wu) at the first time step of the aforementioned cubic cavity run on the $34 \times 34 \times 18$ grid. One working unit is defined as the work equivalent to one iteration on the finest grid. As a base case we use a point SOR method with an optimized over-relaxation parameter of 1.5. In the multigrid method, four grid levels are employed. As expected, the point SOR method converges very slowly. On the other hand, the multigrid method shows a very favorable exponential convergence rate that drops the residual to machine zero in less than 40 working units (16 V-cycles in this case). We note that this convergence rate is achieved on a Cartesian but highly non-uniform grid. Similar behavior has been experienced with curvilinear grids. For mildly skewed grids, the convergence of the multigrid method in solving the pressure Poisson equation remains exponential, although it may become slower as the grid quality continues to deteriorate. Nevertheless, the more irregular a grid is, the more essential it is to use a rapidly converging method such as multigrid.

The CPU and memory usage of the present semi-implicit, non-staggered-grid, non-orthogonal curvilinear code is compared with two staggered-grid codes, one explicit and one fully implicit. These codes employ solution methods similar to those discussed by Perng [10] and were written and tested by the present authors (e.g., see [11]). In the staggered-grid methods, all three Cartesian velocity components are defined on each face of the control volume. The grid layout and variable definition follow that developed by Maliska and Raithby [9]. The explicit staggered-grid code (EXPS) utilizes the second-order Runge–Kutta time-marching method. The fully implicit staggered-grid code (IMPS) employs the trapezoidal scheme and requires global iteration at every time step. Spatial discretization schemes are similar to those used in the present method. The pressure Poisson equations are solved using the same multigrid method in all three codes.

The CPU time (in μs) per cell per time step and the number of variables defined per cell are given in Table I for the three methods. The CPU time was obtained on a Cray Y-MP 8/864 supercomputer. All three codes have been vectorized. Two multigrid V-cycles were performed in solving the pressure Poisson equation at every time step. For IMPS, two global iterations were performed at every time step. We can see that the present semi-implicit code is two times faster than the explicit staggered-grid code and almost four times faster than the implicit staggered-grid code. Moreover, the removal of the viscous stability limit in the semi-implicit method allows for a much larger time step

TABLE I

Comparisons of CPU Time (μs per Cell, per Time Step) and Memory Storage (Number of Variables per Cell) between the Present Method and Previous Staggered Grid Methods

	CPU (μs) per cell per time step	Number of variables per cell
EXPS	60	103
IMPS	110	94
Present	28	51

compared to the explicit code in cases where a very fine grid needs to be used near solid boundaries. Although the time step of the implicit code can be even larger when the goal is to obtain a steady-state result, the step is generally limited by the CFL number in a time-accurate simulation. One reason is the need to control temporal errors. Another reason is that the trapezoidal time-marching scheme produces spurious oscillations if the time step is much larger than the limit set by the CFL condition. The number of iterations that need to be performed every time step in the fully implicit method also increases with time step, making the method less efficient.

As Table I shows, the total number of variables that the present non-staggered-grid method needs to define per control volume is only about half of the two staggered-grid methods. This is attributed to the drastic reduction in the number of metric terms as well as physical variables used in the non-staggered-grid formulation. It is noted that the number of variables in the above two staggered-grid codes EXPS and IMPS has been controlled by careful arrangement. If seven sets of metrics are used as in some previous staggered-grid codes [10, 12], the number of metrics only will be 112 and the number of total variables per control volume will be about 150. The present code executes at a typical speed of 100 MFLOPS on a single processor of a Cray Y-MP 8/864 supercomputer.

4.3. Laminar Flow over a Backward-Facing Step

The laminar flow over a backward-facing step has long been used as a benchmark to examine the accuracy of numerical methods. The flow geometry and the boundary conditions are shown in Fig. 10. The expansion ratio is 1:2. A fully developed parabolic profile is imposed at the inlet, while gradient-free condition is used at the outflow boundary. The length of the computational domain is 30 step heights. A 130×66, streamwise-uniform and vertically nonuniform grid is used in the present computation. In Fig. 11, the computed normalized reattachment length as a function of the Reynolds number is compared with previous experimental and numerical results. The Reynolds number is based on the mean entrance velocity and the step height. Excellent agreement is achieved between the present data

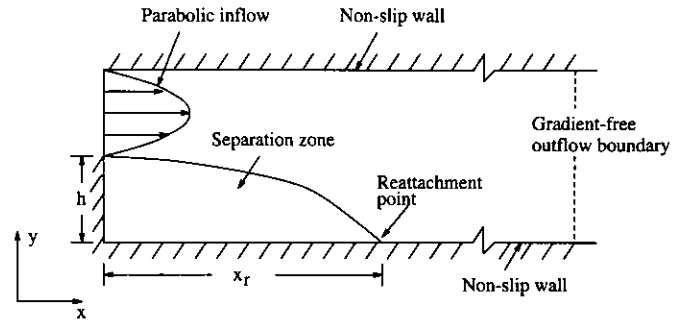


FIG. 10. Geometry and boundary conditions of the flow over a backward-facing step.

and the computation of Kim and Moin [1] in the Reynolds number range from 200 to 800. The discrepancy between the present result and the experimental data of Armaly *et al.* [45] at Reynolds numbers higher than 400 may be due to possible three-dimensional effects in the experiments. Perng's computation [10] was performed with an entrance channel ahead of the expansion. His data predicts smaller reattachment lengths at Reynolds numbers higher than about 300. The discrepancy between the present result and Perng's can be attributed to the upstream influence on the inlet profile. Perng [10] has shown that with an entrance section, the velocity profile right at the expansion deviates from a parabola and appears to have a "down-wash." Nevertheless, the excellent agreement with Kim and Moin's data which was generated with the same boundary condition as used in the present simulations demonstrates the accuracy of the present method.

At higher Reynolds numbers, a separation bubble appears at the top wall of the channel. In Table II, the normalized length of the top separation bubble is compared with the data from [1]. Again, excellent agreement is found.

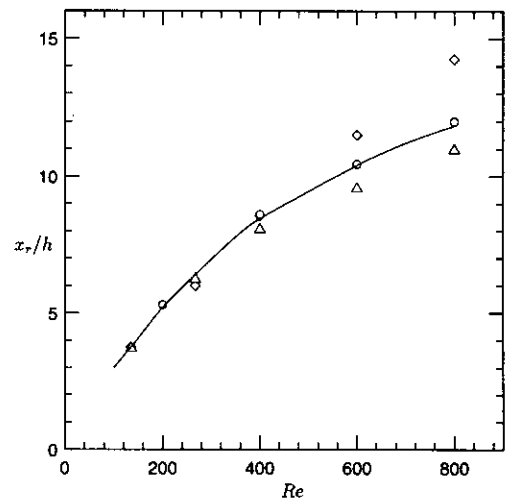


FIG. 11. Reattachment length as a function of the Reynolds number of the flow over a backward-facing step: ◇, Armaly *et al.* [45]; —, Kim and Moin [1]; △: Perng [10]; ○, present.

TABLE II

The Length of the Separation Bubble at the Top Wall of Backward-Facing Step Flows Scaled by the Step Height, x_s/h, vs the Reynolds Number

	Re = 400	Re = 600	Re = 800
Kim & Moin [1]	—	7.8	11.5
Present	2.5	7.8	11.4

4.4. Lid-Driven Flow in a Polar Cavity

The lid-driven flow in a two-dimensional polar cavity is a standard test case for computer codes written in curvilinear coordinates. The geometry and the boundary conditions of the polar cavity together with a 42×42 non-uniform grid are given in Fig. 12. The Reynolds number (based on the lid velocity and the radius of the inner circle) is 350. Figure 13 shows the steady state azimuthal and radial velocity profiles along the line $\theta = 0$. The experimental and numerical results of Fuchs and Tillmark [46] together with the numerical result of Perng [10] are included for comparison. The present result on a uniform 42×42 grid gives a slightly stronger azimuthal flow near the driven lid and a weaker one far from it. There is no discernible difference between the profiles from the present solution on a non-uniform 42×42 grid and those of Perng's data which were obtained also on a 42×42 non-uniform grid. The present calculation also compares well with Fuchs and Tillmark's numerical results. The discrepancy between the numerical and the experimental data may be attributed to the three-dimensional effects in the experiments in which three-dimensional flow structures

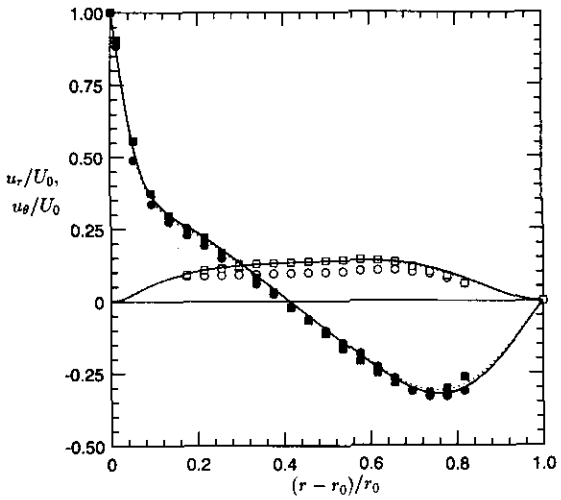


FIG. 13. Comparison of the radial u_r and azimuthal u_θ velocity profiles along the radial line $\theta = 0$ (Re = 350): ●, u_θ Fuchs and Tillmark [46], experiment; ○, u_r [46], experiment; ■, u_θ [46], computation; □, u_r [46], computation; ---, Perng [10]; ···, present 42×42 uniform grid; —, present 42×42 nonuniform grid.

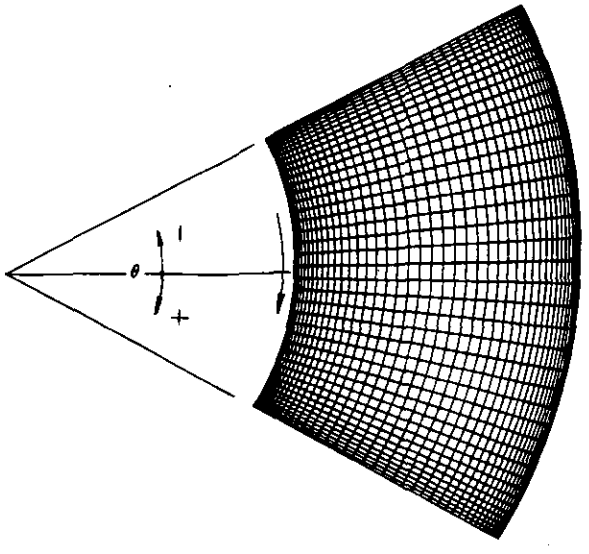


FIG. 12. Geometry and boundary conditions of the lid-driven flow in a polar cavity.

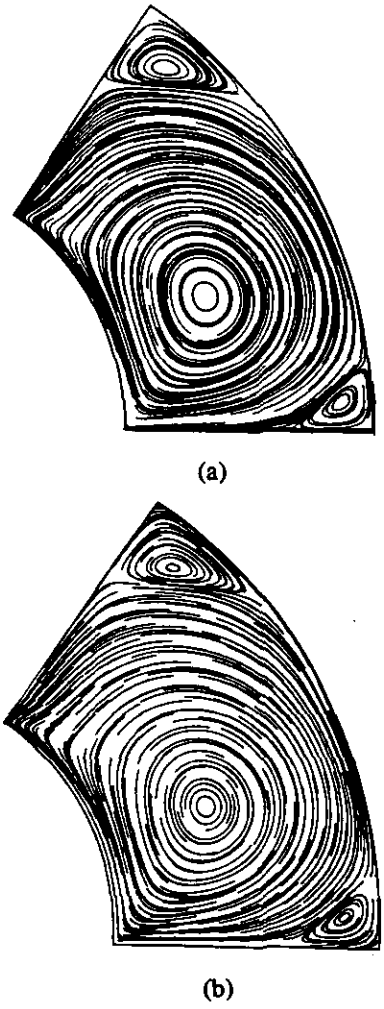


FIG. 14. Comparison of the streamlines between (a) present and (b) Perng [10] (Re = 350).

were observed [46]. Three-dimensional simulations of the original flow in reference [46] are difficult because of the large aspect ratio (about 10:1) of the experimental apparatus employed, and, thus, were not attempted. In Fig. 14, the flowfield at steady state from the present non-uniform grid solution is compared with the result in [10]. Again good agreement is achieved in the size of the vortices and the location of the separation and reattachment points.

4.5. Three-Dimensional Flow in a 90° Bending Square Duct

Several researchers (e.g., [3, 15]) used the three-dimensional flow in a 90° bending square duct as a benchmark to test three-dimensional flow codes written in curvilinear coordinates. The flow structure is complex and truly three-dimensional. The geometry and the coordinate system are shown in Fig. 15. The inlet and the outlet are both five units long. The radius of the bending circle is 1.8 units. Both sides of the square cross section are one unit wide. Due to the symmetry in the z direction, we only compute the flow in a half duct ($0 < z < 0.5$). The fully developed velocity profile in a straight square duct [47] is imposed at the inlet boundary. The gradient-free condition is employed at the outflow boundary. We expect that the upstream influence to the inlet profile is very small in this case due to the long inlet section. The Reynolds number based on the mean entrance velocity and the duct width is 790. Two runs were carried out. One used a non-uniform $34 \times 18 \times 10$ grid and the other used a non-uniform $66 \times 34 \times 18$ grid. Grid points are uniformly distributed in the streamwise direction and clustered near the solid walls. In Fig. 16, the steady-state streamwise velocities at six streamwise stations and two spanwise z locations are plotted. Good agreement is achieved between the present results and the experimental measurements of Humphrey *et al.* [48] and the numerical solutions of Rosenfeld *et al.* [15]. The numerical method employed by Rosenfeld *et al.* is also based on a fractional step method with the volume flux being used as the primary

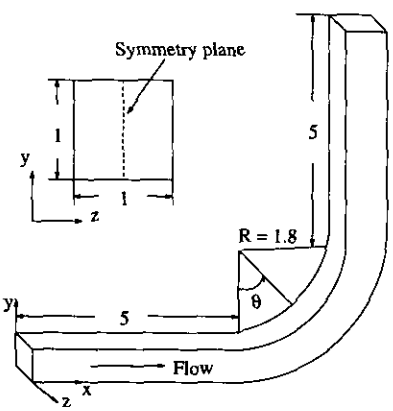


FIG. 15. Geometry of the 90° bending duct.

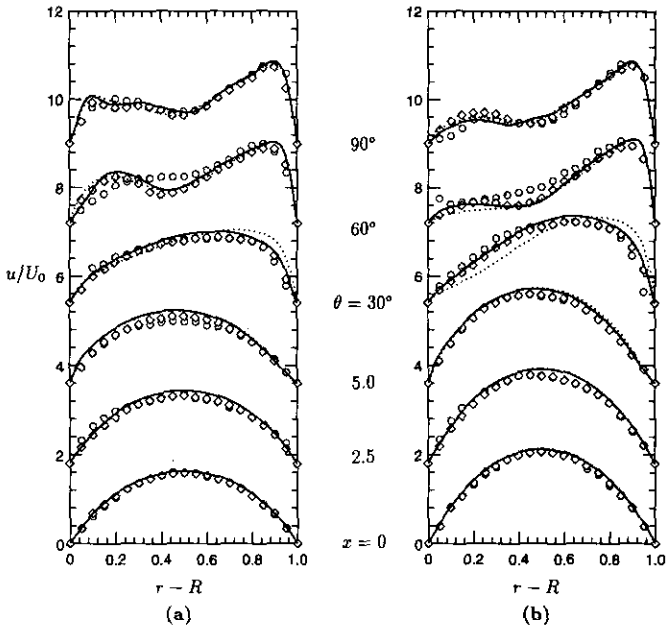


FIG. 16. Comparison of the streamwise velocity profiles at six streamwise stations along (a) $z = 0.25$; (b) $z = 0.5$ (Re = 790): ○, Humphrey *et al.* [48]; ◇, Rosenfeld *et al.* [15]; ..., present $34 \times 18 \times 10$ grid; —, present $66 \times 34 \times 18$ grid.

dependent variable. Their solution given in Fig. 16 is obtained on a $60 \times 40 \times 20$ nonuniform grid. The present solutions slightly overpredict the streamwise velocity at the first three stations downstream of the inlet. However, they accurately capture the peaks near the outside wall at the last two downstream stations. The fine grid solution exhibits better agreement with the reference experimental and

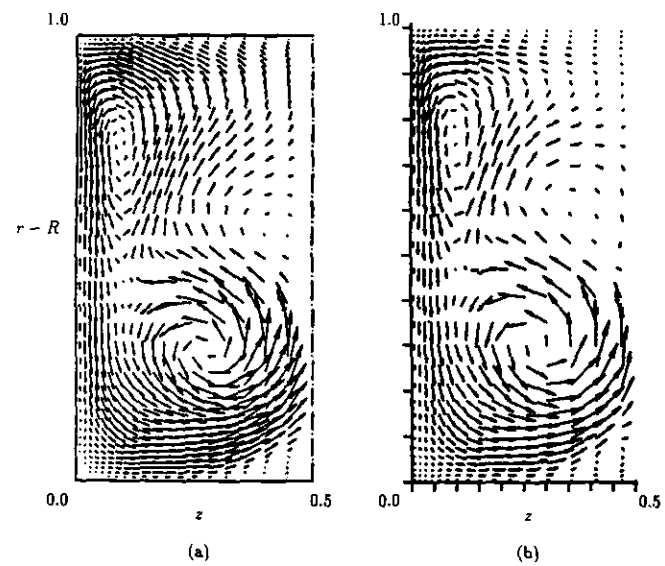


FIG. 17. Comparison of the velocity field in the cross section at $\theta = 90°$ between (a) Rosenfeld *et al.* [15] and (b) present ($66 \times 34 \times 18$ grid).

numerical data. To investigate the effect of the outflow location, a run was performed in which the length of the outlet after the bend was doubled and a $98 \times 34 \times 18$ grid was used. No discernible difference was found between the two present fine grid solutions. The discrepancy between the experimental and both of the numerical results requires further investigation. Nevertheless, the results validate the present method's capability to accurately simulate complex three-dimensional flows.

In Fig. 17, the cross-sectional velocity vector field at $\theta = 90°$ is plotted and compared with the result from [15]. Good qualitative agreement is found. We can see that the flow exhibits complex vortical structures when passing through the bend.

5. UNSTEADY AND COMPLEX-FLOW CAPABILITY

One of the major purposes of the present work is to develop an overall efficient and accurate numerical method for both physically and geometrically complex flows. We demonstrate this capability by presenting the large eddy simulation results of a turbulent stratified and rotating flow in an irregular domain which is a laboratory model of the coastal upwelling phenomenon. To the author's knowledge, this is the first attempt of performing a large eddy simulation of this flow.

5.1. Upwelling Flow Description

The geometry of the flow domain is a 90°-section of an annulus with a sloping bottom. A schematic of the domain and the upwelled density interface are shown in Fig. 18. Initially, the stratified fluid is in solid body rotation with the container. The initial density field which is horizontally uniform and vertically separated by a narrow interface approximates a "two-layer" stratified system with a lighter fluid on top of a heavier fluid. At time $t = 0$, upwelling flow is generated by relative rotation of the top lid, which simulates a wind stress (Fig. 18). This simulation was chosen because measurement data of upwelling flows in a rotating cylinder was available for comparison [49]. The geometric parameters which are given in Table III are similar to those in [49] except that, instead of a cylinder, a section of an annulus is used. The outer radius of the annulus R_0 is 0.45 m and the inner radius R_1 is 0.1 R_0. The effect of the presence of the inner wall on the upwelling flow has been shown to be insignificant [24].

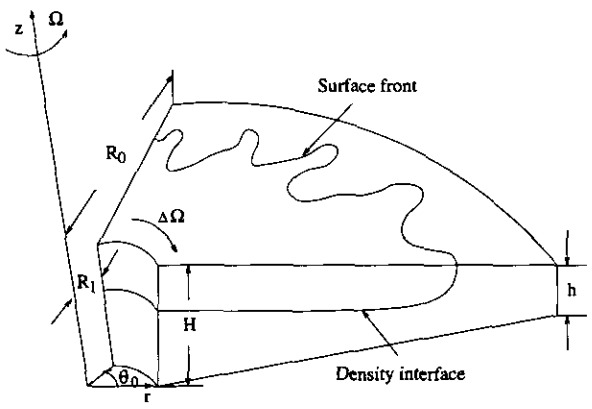


FIG. 18. Schematic of the flow domain and the density interface of the upwelling flow.

TABLE III

The Geometric and Physical Parameters of the Upwelling Flow Simulation

θ_0 (rad)	H (m)	h (m)	Ω (s^{-1})	$\Delta\Omega$ (s^{-1})	h_0 (cm)	$\delta\rho/\rho$
$\pi/2$	0.15	0.033	2.27	−0.185	2.60	0.018

The governing equations are the grid-filtered continuity, Navier–Stokes, and the scalar transport equations with the Boussinesq approximation which are solved in the reference frame rotating with the container. The large scale structures are resolved by the simulation while the small scale motion is represented by the dynamic eddy viscosity subgrid-scale model [41]. For details of the governing equations and the subgrid-scale model, the reader is referred to [24].

A no-slip condition for the velocity is applied to the top, bottom, inner, and outer walls. A no-flux condition is used for the density at the solid walls. At the two azimuthal boundaries, periodic boundary conditions are applied. A $66 \times 66 \times 66$ grid is employed which is non-uniform in the radial and vertical direction but uniform in the azimuthal direction. Grid points are clustered in the vicinity of solid walls. Geometric stretching is employed in the directions where grid distribution is non-uniform. The spatial and temporal resolutions are shown in Table IV.

The fluid is salt stratified with a Prandtl number of 723. The physical parameters are given in Table III. The physical conditions used in the simulation are the same as in Case (a) of [49] except that the fluid viscosity is 12.5 times larger than that in the experiment. This larger viscosity is necessary because the simulation of the original experiment has been shown to be computationally infeasible [24]. Since the upwelling phenomenon under investigation is

TABLE IV

The Spatial and Temporal Resolution of the Upwelling Flow Simulation

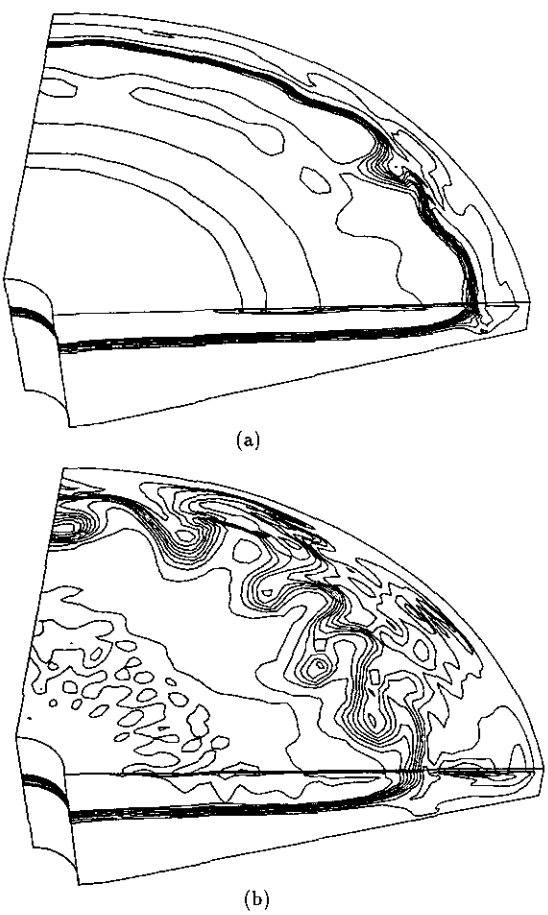
Grid resolution ($N_r \times N_\theta \times N_z$)	$\frac{\Delta r_{\min}}{(R_0 - R_1)}$	$\frac{\Delta z_{\min}}{H}$	Time step Δt (s)
$66 \times 66 \times 66$	3×10^{-3}	3×10^{-3}	2×10^{-2}

essentially inviscid, the increase of the fluid viscosity does not have a significant effect. This is confirmed by the agreement between the computed and the measured data (see the following section and [24]).

5.2. Simulation Results

As the lid rotates clockwise relative to the system rotation, fluid near the top which moves with the lid is driven radially inward by the Coriolis force and forms the top Ekman layer. Constrained by mass conservation, fluid at the bottom moves upward which lifts up the density interface. If the relative rotation of the top lid is strong enough, the body of the interface will intersect the surface and form a surface front. During this process, unstable stratification occurs as the heavy fluid reaches the top surface and intensive turbulent mixing exists in the vicinity of the front.

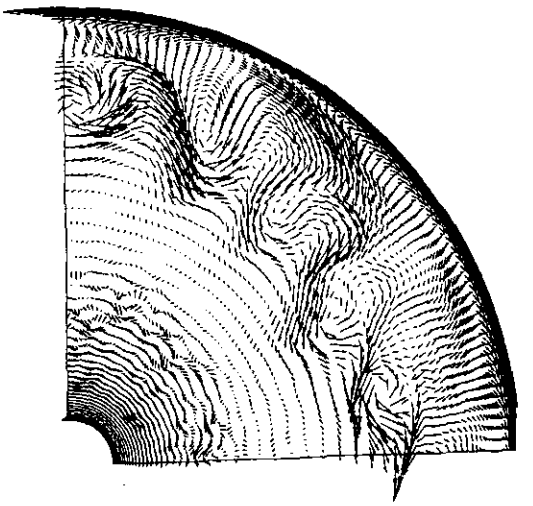
The surface front which initially is axisymmetric undergoes baroclinic instability. Azimuthal waves appear at the front and grow to large amplitude. Figure 19a shows the density field at $t/t_s = 0.9$, where t_s is the spin-up time scale which is defined as $t_s = (h_{10}/\Delta\Omega)((\Omega + \Delta\Omega)/\nu)^{1/2}$ and h_{10} is the initial depth of the top layer. The front appears to be almost axisymmetric although small amplitude azimuthal waves are evident. Figure 19b shows the density field at $t/t_s = 2.0$. The initially axisymmetric front has evolved into an irregular pattern with large-scale frontal waves and eddies. The front is wider due to mixing on both sides. Figure 20 compares the simulated and the observed structures of the frontal baroclinic eddies in a horizontal plane. We see that the simulated structures are very similar to those observed in the flow visualizations.



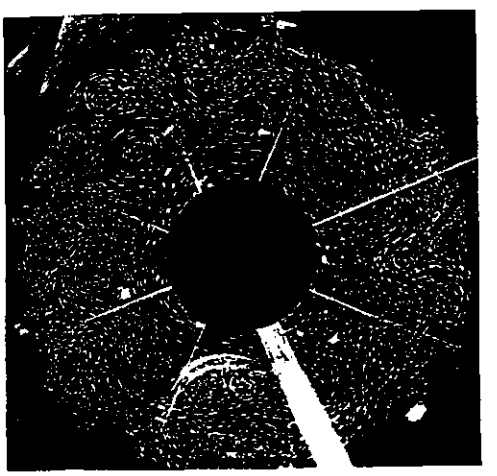
(a)

(b)

FIG. 19. Density field of the upwelling flow: (a) $t/t_s = 0.9$; (b) $t/t_s = 2.0$.



(a)



(b)

FIG. 20. Comparison of the structures of the frontal waves in a reference frame traveling with the waves: (a) simulated velocity field; (b) flow visualization [49].

The frontal azimuthal waves become saturated after a certain time and the growth stops. The size of the saturated waves is measured by the dimensionless wavelength $\hat{L}_w = L_w/2\pi\hat{R}$, where $\hat{R} = (\bar{R}_1\bar{R}_2)^{1/2}$, $\bar{R}_i\,[=(g'h_i)^{1/2}/f(i=1,2)]$ is

the Rossby deformation radius of the ith layer with the initial depth of h_i, $g' = g\,\delta\rho/\rho$ is the reduced gravity, and $f = 2\Omega$ is the Coriolis parameter. The computed $\hat{L}_w$ is 1.00 while the measured value is 0.98 [49]. More quantitative comparisons can be found in [24].

6. SUMMARY

A numerical method to solve the time-dependent, three-dimensional incompressible Navier–Stokes equations in curvilinear coordinates is presented. A non-staggered grid is employed with the pressure and the Cartesian velocity components being defined at cell centers and the volume fluxes being defined on their corresponding faces. Only one set of metric quantities needs to be stored per control volume and no averaging of metrics is required.

The non-staggered grid is naturally combined with the fractional step method. The momentum equations are advanced semi-implicitly using the approximate factorization technique. A third-order accurate upwind quadratic scheme is used to interpolate the intermediate velocity onto the cell faces. The pressure Poisson equation is formulated in the same way as in a staggered-grid method and is solved iteratively by a multigrid method. Mass conservation can be satisfied to machine accuracy. The boundary condition for pressure is obtained from the momentum equations. The method is second-order accurate in both space and time, which is verified by comparing the numerical solution with an analytical solution.

Solutions of benchmark cases were compared with existing analytical, numerical, and experimental data. Good agreement was found in all cases. The CPU time and memory storage were significantly reduced compared with previous staggered-grid codes. Results of the large eddy simulation of a turbulent, stratified, and rotating upwelling flow in an irregular container demonstrated the unsteady and complex-flow capability of the present method.

ACKNOWLEDGMENTS

The authors thank Professor J. Ferziger, Professor J. Oliger, and Dr. C.-Y. Perng for very helpful discussions. The research was supported by National Science Foundation Grant MSM-8719509. Part of the computations was performed on Cray Y-MP 8/864 at San Diego Supercomputer Center and National Center of Atmospheric Research which are sponsored by the National Science Foundation.

REFERENCES

1. J. Kim and P. Moin, *J. Comput. Phys.* **59**, 308 (1985).
2. S. V. Patankar, "Numerical Heat Transfer and Fluid Flow" (Hemisphere, New York, 1980).
3. S. E. Rogers, D. Kwak, and C. Kiris, AIAA Paper 89-0463, 1989 (unpublished).
4. A. J. Chorin, *J. Comput. Phys.* **2**, 12 (1967).
5. J. L. Steger and P. Kutler, *AIAA J.* **15**, 581 (1977).
6. A. J. Chorin, *Math. Comput.* **22**, 745 (1968).
7. R. Temam, *Arch. Rat. Mech. Anal.* **32**, 377 (1969).
8. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8**, 2182 (1965).
9. C. R. Maliska and G. D. Raithby, *Int. J. Numer. Methods Fluids* **4**, 519 (1984).
10. C.-Y. Perng, Ph.D. dissertation, Dept. Mech. Eng., Stanford University, 1990 (unpublished).
11. Y. Zang, R. L. Street, and J. R. Koseff, in *Proceedings, Seventh International Conference on Numerical Methods in Laminar and Turbulent Flow, Stanford, CA*, July 11–15, 1991, edited by C. Taylor *et al.*, p. 1485.
12. R. L. Meakin and R. L. Street, *Comput. Methods Appl. Mech. Eng.* **68**, 151 (1988).
13. W. Shyy and T. C. Vu, *J. Comput. Phys.* **92**, 82 (1991).
14. R. F. Van Der Wijngaart, Ph.D. dissertation, Dept. Mech. Eng., Stanford University, 1989 (unpublished).
15. M. Rosenfeld, D. Kwak, and M. Vinokur, *J. Comput. Phys.* **94**, 102 (1991).
16. K. C. Karki and S. V. Patankar, *AIAA J.* **27**, 1167 (1989).
17. S. Abdallah, *J. Comput. Phys.* **70**, 182 (1987).
18. M. L. Mansour and A. Hamed, *J. Comput. Phys.* **86**, 147 (1990).
19. C. M. Rhie and W. L. Chow, *AIAA J.* **21**, 1525 (1983).
20. M. Perić, R. Kessler, and G. Scheuerer, *Comput. Fluids* **16**, 389 (1988).
21. S. Majumdar, *Numer. Heat Transfer* **13**, 125 (1988).
22. W. Rodi, S. Majumdar, and B. Schönung, *Comput. Methods Appl. Mech. Eng.* **75**, 369 (1989).
23. S. W. Armfield, *Comput. Fluids* **20**, 1 (1991).
24. Y. Zang, Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, 1993 (unpublished).
25. B. P. Leonard, *Comput. Methods Appl. Mech. Eng.* **19**, 59 (1979).
26. C.-Y. Perng and R. L. Street, *Int. J. Numer. Methods Fluids* **9**, 341 (1989).
27. G. D. Raithby, *Comput. Methods Appl. Mech. Eng.* **9**, 153 (1976).
28. M. A. Leschziner, *Comput. Methods Appl. Mech. Eng.* **23**, 293 (1980).
29. T. Han, J. A. C. Humphrey, and B. E. Launder, *Comput. Methods Appl. Mech. Eng.* **29**, 81 (1981).
30. A. Pollard and A. L. W. Siu, *Comput. Methods Appl. Mech. Eng.* **35**, 293 (1982).
31. H. Koo and S. O. Park, *Commun. Appl. Numer. Methods* **7**, 111 (1991).
32. T. Hayase, J. A. C. Humphrey, and R. Greif, *J. Comput. Phys.* **98**, 108 (1992).
33. B. P. Leonard, in *AIAA/ASME/SIAM/APS First National Fluid Dynamics Congress, 1988*, p. 226.
34. R. M. Beam and R. F. Warming, *J. Comput. Phys.* **22**, 87 (1976).
35. W. R. Briley and H. McDonald, *J. Comput. Phys.* **24**, 428 (1977).
36. A. Brandt, *Math. Comput.* **31**, 333 (1977).
37. U. Ghia, K. N. Ghia, and C. T. Shin, *J. Comput. Phys.* **48**, 387 (1982).
38. A. K. Prasad, C.-Y. Perng, and J. R. Koseff, in *First National Fluid Dynamics Congress, Cincinnati, Ohio, 1988.*
39. J. R. Koseff, R. I. Street, P. M. Gresho, C. D. Upson, J. A. C. Humphrey, and W.-M. To, in *Proceedings, Third International Conference on Numerical Methods in Laminar and Turbulent Flows, Seattle, Wash., Aug. 8–11, 1983*, edited by C. Taylor, p. 564.
40. A. K. Prasad, Ph.D. dissertation, Dept. Mech. Eng., Stanford University, 1989 (unpublished).

41. M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, *Phys. Fluids A* **3**, 1760 (1991).
42. Y. Zang, R. L. Street, and J. R. Koseff, in *ASME Fluids Engineering Conference, Washington, DC, 1993.*
43. Y. Zang, R. L. Street, and J. R. Koseff, *Phys. Fluids A* **5**, 3186 (1993).
44. A. K. Prasad and J. R. Koseff, *Phys. Fluids A* **1**, 208 (1988).
45. B. F. Armaly, F. Durst, and J. C. F. Pereira, *J. Fluid Mech.* **127**, 473 (1983).
46. L. Fuchs and N. Tillmark, *Int. J. Numer. Methods Fluids* **5**, 311 (1985).
47. F. M. White, *Viscous Fluid Flow* (McGraw–Hill, New York, 1974), p. 123.
48. J. A. C. Humphrey, A. M. K. Taylor, and J. H. Whitelaw, *J. Fluid Mech.* **83**(3), 509 (1977).
49. S. Narimousa, T. M. Maxworthy, and G. R. Spedding, *J. Fluid Mech.* **223**, 113 (1991).