AIAA JOURNAL
Vol. 32, No. 8, August 1994

Finite Element Method for Incompressible Viscous Flows with Adaptive Hybrid Grids

Y. Kallinderis* and K. Nakajima†
University of Texas at Austin, Austin, Texas 78712

A new adaptive finite element numerical method has been developed for the unsteady Navier-Stokes equations of incompressible flow in two dimensions. The momentum equations combined with a pressure correction equation are solved employing a nonstaggered grid. The solution is advanced in time with an explicit/implicit marching scheme. An adaptive algorithm has been implemented, which refines the grid locally to resolve detected flow features. A combination of quadrilateral as well as triangular cells provides a stable and accurate numerical treatment of grid interfaces that are located within regions of high gradients. Applications of the developed adaptive algorithm include both steady and unsteady flows, with low and high Reynolds numbers. Comparisons with analytical as well as experimental data evaluate accuracy and robustness of the method.

I. Introduction

INCOMPRESSIBLE flows are frequently encountered in engineering applications. During the past two decades a significant number of numerical algorithms have been developed for solution of the incompressible Navier-Stokes equations.[1] The lack of pressure term in the continuity equation makes solution of the momentum equations with the divergence-free constraint more difficult. In the case of incompressible flows, the conservation of mass acts as a constraint condition that the velocity field must satisfy, whereas in compressible flows, the conservation of mass is given through a partial differential equation for the temporal variation of density. The infinite speed of sound in the incompressible case requires an implicit treatment of the pressure. Furthermore, spatial discretization for pressure and velocity may produce oscillatory solutions.

One approach followed is to formulate the equations in terms of a stream function and a vorticity. Extension of this method to three dimensions is not possible. Different formulations have been used in three dimensions, such as the vorticity-velocity approach. Another approach is to use the compressible flow equations and solve them for low Mach numbers. The required time step for such computations is very small because the speed of sound approaches infinity at the incompressible limit. A method that uses compressible-like governing equations is the artificial compressibility approach.[2–4] A time derivative of the pressure is added to the continuity equation, and the incompressible flowfield is treated as compressible during the transient stage. Time accuracy of the simulation is usually not preserved.

Another class of algorithms uses a special Poisson equation for the pressure field.[5–8] The usual computational procedure is to assume an initial pressure field, and then an iterative process is defined until the continuity equation is satisfied. A major issue of the corresponding pressure and velocity spatial discretization is oscillations in the pressure field. To reject these modes, staggered grids have been employed by several of these algorithms.[9,10] On the other hand, employment of nonstaggered grids[11–13] requires dissipation in the algorithms. Stability of both approaches with high-Reynolds-number flows is an important issue. A review of numerical methods for incompressible flows, as well as references of previous work, are given in Ref. 1. Furthermore, extensive literature on finite element methods is given in Ref. 14.

Resolution of the computational mesh plays a crucial role in accuracy of computations. However, generation of a grid that both fits the flow geometry and resolves the local flow features is quite difficult and even impossible in some cases. In general, the selection of the grid that is to be used in a numerical simulation is determined a priori before starting the solution procedure, and quite often the grid is modified by the user to improve the results. Adaptive grid algorithms are flexible enough to adjust the grid during the solution procedure without intervention by the user. Frequently, the regions that require high resolution are very small compared with the size of the overall computational domain. Local grid embedding consists of division of cells to reduce the truncation error and to have a more equal distribution of it throughout the solution domain. Quadrilateral meshes have been employed for inviscid compressible flows[15,16] as well as for turbulent compressible flows.[17,18]

One of the most serious problems with using quadrilaterals has been the presence of grid-interfaces, which require special numerical treatment,[19] which can be quite complicated. Triangular meshes have also been employed with grid embedding for compressible flows.[16,20] An attractive feature of unstructured grids is their ability to eliminate grid interfaces. The flow solver requires no further modifications when employing such an adapted grid. However, quadrilaterals are more suitable for solving boundary layers, which require very thin grid cells because of the strong directional gradients. Finally, staggered grids have been quite popular with incompressible flow solvers, but they greatly complicate the treatment of grid interfaces when local embedding is used.

In the present work, an adaptive finite element numerical scheme has been developed for the unsteady Navier-Stokes equations of incompressible flow in two dimensions. A combination of quadrilateral as well as triangular cells provides flexibility in forming the adaptive grids. Thin quadrilateral elements resolve boundary layers, while triangles eliminate special interface (hanging) nodes. The border regions between the two types of elements are placed within regions of high flow gradients to test stability and accuracy of the developed formulation. Comparisons with the results obtained with equivalent globally fine grids that contain no interfaces are also employed. The momentum equations combined with a pressure correction equation are solved employing a nonstaggered grid where all of the dependent variables are defined at the cell corners. The solution is advanced in time with an explicit/implicit marching scheme. An adaptive algorithm has been implemented, which refines the grid locally to resolve detected flow features. Employment of a nonstaggered grid facilitates application of adaptive gridding. Applications of the developed adaptive algo-

Received May 11, 1993; revision received Dec. 30, 1993; accepted for publication Feb. 21, 1994.

*Associate Professor, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

†Graduate Research Assistant; currently Senior Research Associate, Saftey Engineering Department, Mitsubishi Research Institute, Inc., 2-3-6 Otemachi, Chiyoda-ku, Tokyo 100, Japan.

rithm include both steady and unsteady flow, as well as flow of high Reynolds number.

In the following, the pressure correction formulation and the finite element discretization are presented. Next, the adaptive hybrid-grid algorithm is described. Finally, steady as well as unsteady flow simulations are presented, and comparisons with analytical and experimental data are performed.

II. Governing Equations and Pressure Correction Formulation

The governing equations are the following nondimensionalized continuity and Navier-Stokes equations of incompressible flow:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) = -\nabla p + \frac{1}{Re}\Delta \iota \tag{2}$$

An explicit/implicit marching scheme is adopted for integration in time of the previous equations. The velocity values are treated explicitly, whereas the pressure values are treated implicitly in the momentum equations.[1] The velocity values are marched in time with a forward Euler scheme.[5] The continuity equation is formulated implicitly with the velocity values considered at time level $(n+1)$. Specifically, the corresponding semidiscrete system is written as follows:

$$\nabla \cdot \boldsymbol{u}^{(n+1)} = 0 \tag{3}$$

$$\frac{\boldsymbol{u}'^{(n+1)} - \boldsymbol{u}^{(n)}}{\Delta t} + \nabla \cdot [\boldsymbol{u}^{(n)}\boldsymbol{u}^{(n)}] = -\nabla p^{(n+1)} + \frac{1}{Re}\Delta \boldsymbol{u}^{(n)} \tag{4}$$

where the superscripts denote the time levels. The previous equation cannot be solved directly due to the implicit treatment of the pressure term. An auxiliary velocity vector $\boldsymbol{u}'$ is introduced, which satisfies the following equation:

$$\frac{\boldsymbol{u}' - \boldsymbol{u}^{(n)}}{\Delta t} + \nabla \cdot [\boldsymbol{u}^{(n)}\boldsymbol{u}^{(n)}] = -\nabla p^{(n)} + \frac{1}{Re}\Delta \boldsymbol{u}^{(n)} \tag{5}$$

In this equation, the pressure term is treated explicitly and $\boldsymbol{u}'$ can be obtained directly. However, the solution $\boldsymbol{u}'$ does not satisfy the continuity equation. Subtracting Eq. (5) from Eq. (4), one obtains:

$$\boldsymbol{u}^{(n+1)} - \boldsymbol{u}' = -\{\nabla [p^{(n+1)} - p^{(n)}]\}\,\Delta t \tag{6}$$

Introducing a scalar potential ϕ, such that

$$\boldsymbol{u}^{(n+1)} - \boldsymbol{u}' = -\nabla\phi \tag{7}$$

the following equation for pressure can be obtained:

$$p^{(n+1)} - p^{(n)} = \frac{1}{\Delta t}\phi \tag{8}$$

Finally, taking the divergence of each side of Eq. (7) and considering the continuity equation [Eq. (3)], the following pressure correction Poisson equation is obtained:

$$\Delta\phi = \nabla \cdot \boldsymbol{u}' \tag{9}$$

In this equation, the values of ϕ on the left-hand side are treated implicitly, which requires inversion of a matrix. Using the ϕ values obtained by the previous equation, we can correct the velocity and pressure fields using Eqs. (7) and (8) as follows:

$$\boldsymbol{u}^{(n+1)} = \boldsymbol{u}' - \nabla\phi \tag{10}$$

$$p^{(n+1)} = p^{(n)} + \frac{1}{\Delta t}\phi \tag{11}$$

The previous solution procedure follows the explicit/implicit marching scheme in Ref. 8. The overall solution procedure corresponding to marching by one time step is summarized as follows:

1) Calculate the auxiliary velocity vector $\boldsymbol{u}'$ from Eq. (5) using $\boldsymbol{u}^{(n)}$ and $p^{(n)}$ values.

2) Solve Eq. (9) and obtain the ϕ values.

3) Calculate $\boldsymbol{u}^{(n+1)}$ and $p^{(n+1)}$ using Eqs. (10) and (11).

4) If $\nabla \cdot \boldsymbol{u}^{(n+1)} < \varepsilon$ where ε is the tolerance for divergence, advance to the next time step. If not, consider $\boldsymbol{u}^{(n+1)}$ as $\boldsymbol{u}'$ and repeat steps 2 and 3.

III. Finite Element Discretization

Both momentum and pressure correction equations are discretized using the Galerkin finite element approach on a nonstaggered grid. The scheme is compact with all operations being restricted to within each grid cell. Linear triangles and bilinear isoparametric quadrilateral elements are employed.[10]

A. Momentum Equations

The momentum equation [Eq. (5)] can be written as

$$\begin{aligned} u' &= u^{(n)} - \Delta t\left\{(uu)^{(n)}_{,x} + (uv)^{(n)}_{,y} + p^{(n)}_{,x} - \frac{1}{Re}[u^{(n)}_{,xx} + u^{(n)}_{,yy}]\right\} \\ v' &= v^{(n)} - \Delta t\left\{(uv)^{(n)}_{,x} + (vv)^{(n)}_{,y} + p^{(n)}_{,y} - \frac{1}{Re}[v^{(n)}_{,xx} + v^{(n)}_{,yy}]\right\} \end{aligned} \tag{12}$$

where the subscripts $,x$, $,y$, $,xx$, and $,yy$ denote the operations for partial differentiation.

For quadrilaterals, the values of u, v, and p are defined in each element using the following finite element formulation:

$$u = \sum_{i=1}^{4} N_i u_i, \qquad v = \sum_{i=1}^{4} N_i v_i, \qquad p = \sum_{i=1}^{4} N_i p_i \tag{13}$$

where u_i, v_i, and p_i are nodal values of u, v, and p, and N_i is the interpolating shape function associated with the ith node. In the present method, u, v, and p are defined at cell vertices.

Substituting Eq. (13) into Eq. (12), integrating over each element domain Ω using the Galerkin method, and then considering the Gauss theorem and boundary conditions, we can get the following equations for each element[10]:

$$\begin{aligned} M_{ij}u'_j &= M_{ij}u^{(n)}_j - \Delta t\,\{K^x_{ij}[u^{(n)}_j u^{(n)}_j + p^{(n)}_j] + K^y_{ij}u^{(n)}_j v^{(n)}_j\} \\ &\quad - \frac{\Delta t}{Re}D_{ij}u^{(n)}_j \\ M_{ij}v'_j &= M_{ij}v^{(n)}_j - \Delta t\,\{K^x_{ij}u^{(n)}_j v^{(n)}_j + K^y_{ij}[v^{(n)}_j v^{(n)}_j + p^{(n)}_j]\} \\ &\quad - \frac{\Delta t}{Re}D_{ij}v^{(n)}_j \end{aligned} \tag{14}$$

where

$$M_{ij} = \iint_\Omega N_i N_j \,\mathrm{d}x\,\mathrm{d}y$$

$$K^x_{ij} = \iint_\Omega N_i \frac{\partial N_j}{\partial x}\,\mathrm{d}x\,\mathrm{d}y, \qquad K^y_{ij} = \iint_\Omega N_i \frac{\partial N_j}{\partial y}\,\mathrm{d}x\,\mathrm{d}y$$

$$D_{ij} = \iint_\Omega \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right)\mathrm{d}x\,\mathrm{d}y$$

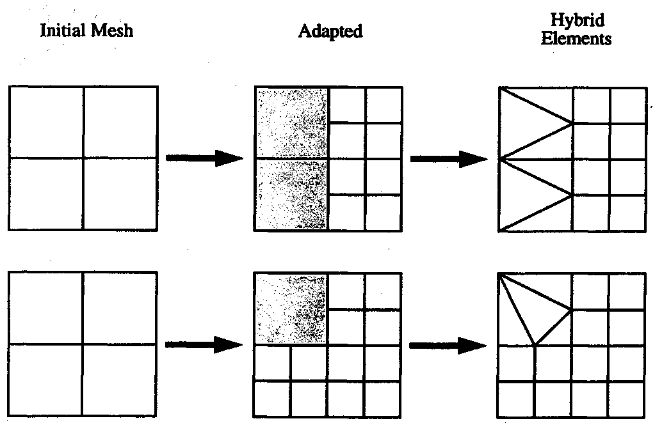

Fig. 1 Elimination of interface nodes by employing triangular elements.

We can construct the following global matrix by assembling the element matrix obtained in Eq. (14):

$$Mu' = Mu^{(n)} - \Delta t F_u$$
$$Mv' = Mv^{(n)} - \Delta t F_v \tag{15}$$

where

$$M = \sum_e M_{ij}, \qquad u^T = [u_1, u_2, u_3, \ldots], \qquad v^T = [v_1, v_2, v_3, \ldots]$$

$$F_u = \sum_e \left\{ K_{ij}^x [u_i^{(n)} u_j^{(n)} + p_j^{(n)}] + K_{ij}^y u_i^{(n)} v_j^{(n)} + \frac{1}{Re} D_{ij} u_j^{(n)} \right\}$$

$$F_v = \sum_e \left\{ K_{ij}^x u_i^{(n)} v_j^{(n)} + K_{ij}^y [v_i^{(n)} v_j^{(n)} + p_j^{(n)}] + \frac{1}{Re} D_{ij} v_j^{(n)} \right\}$$

where the summation is all of the elements. The consistent mass matrix M is used. Therefore, u' and v' are obtained as follows:

$$u' = u^{(n)} - \Delta t M^{-1} F_u$$
$$v' = v^{(n)} - \Delta t M^{-1} F_v \tag{16}$$

B. Pressure Correction Equation

The Poisson equation for pressure correction is solved with the finite element approach. The same type of bilinear quadrilateral element is used as for the momentum equations. The values of u, v, and ϕ are defined in each element using the same expression as in Eq. (13). In the present method, u, v, and ϕ are defined at cell nodes. Integrating Eq. (9) over each element domain Ω using the Galerkin method, one obtains the following equation:

$$\iint_\Omega N_i (\phi_{,xx} + \phi_{,yy})\, dx\, dy = \iint_\Omega N_i (u'_{,x} + v'_{,y})\, dx\, dy \tag{17}$$

Applying Gauss's theorem, we can get the following element matrix system:

$$D_{ij}\phi_j = K_{ij}^x u_j' + K_{ij}^y v_j' \tag{18}$$

where

$$K_{ij}^x = \iint_\Omega N_i \frac{\partial N_j}{\partial x} dx\, dy, \qquad K_{ij}^y = \iint_\Omega N_i \frac{\partial N_j}{\partial y} dx\, dy$$

$$D_{ij} = -\iint_\Omega \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) dx\, dy$$

We can construct the following global matrix system by assembling the element matrix obtained in Eq. (18):

$$D\Phi = f \tag{19}$$

where

$$D = \sum_e D_{ij}, \qquad \Phi^T = [\phi_1, \phi_2, \phi_3, \ldots]$$

$$f = \sum_e [K_{ij}^x u_j' + K_{ij}^y v_j']$$

The values for ϕ on the left-hand side of Eq. (19) are treated implicitly, which requires solution of a system. The matrix that requires inversion is a symmetric linear matrix. The system is solved by the incomplete Cholesky conjugate gradient (ICCG) iterative method.[21] The rate of convergence of the iterative process depends on the numbering of nodes. To accelerate convergence, the elements of the matrix are renumbered using the reverse Cuthill-Mckee (RCM) method.[21] Finally, the velocity and pressure values are obtained by integrating Eqs. (10) and (11).

C. Artificial Dissipation

Central space differencing schemes are susceptible to oscillatory modes in the velocity field of high-Reynolds-number flows. Furthermore, odd-even decoupling of the solution may appear in the pressure field for this nonstaggered type of mesh that is employed.

In the present work, a fourth-order smoothing term is added explicitly to the momentum equations to suppress odd-even decoupling of the velocity solution.[22] Furthermore, fourth-order dissipation is added to the pressure correction equation to stabilize the solution and suppress oscillations in the pressure field.

The smoothing operator is cast in a form suitable for adaptive unstructured grids. All operations are split in such a way that no information is required from outside of each cell. Each grid node receives contributions from each one of its surrounding cells. The operator is formed in two steps. The second-order difference operator is formed in the first step. The second-order distributions to cell corners (j) for the momentum equations are as follows:

$$D_j^2(u) = \left(\sum_{i=1}^{4} u_i \right) - 4u_j \tag{20}$$

The second step duplicates the first, replacing state variables by second-order differences from the first step. The fourth-order smoothing distributions are

$$-D_j^4(u) = \left[\sum_{i=1}^{4} D_i^2(u) \right] - 4D_j^2(u) \tag{21}$$

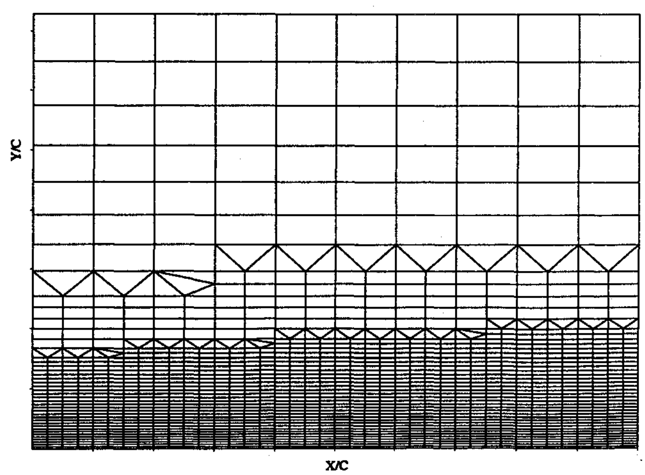

Fig. 2 Two-level adapted grid for Blasius boundary layer with $Re = 10^3$, 1530 points (initial grid has 11 × 19 points).

It should be noted that similar edge-based operations of Eqs. (20) and (21) are applied to triangular meshes as well.

The fourth-order difference terms of u and v are multiplied by empirical coefficients $\sigma_4(u)$ and $\sigma_4(v)$. A large value of σ_4 stabilizes the solution but destroys the accuracy. Therefore, special care is required for choosing the value of σ_4. Numerical experiments have been carried out to determine optimum values for the smoothing coefficient. The determined values are such that the solution accuracy is not affected, whereas the odd-even modes are suppressed.[23]

Similarly, fourth-order difference terms of pressure are added to the right-hand side of the Poisson equation [Eq. (19)] as follows:

$$\boldsymbol{D\Phi} = \boldsymbol{f} + \boldsymbol{d} \tag{22}$$

where

$$\boldsymbol{d}^T = \{\sigma_4(p) D_j^4 [\boldsymbol{p}_j^{(n)}]\}$$

Generally, the value of $\sigma_4(p)$ is different from $\sigma_4(u)$ and $\sigma_4(v)$. In the present work, the values of $\sigma_4(u)$ and $\sigma_4(v)$ are the same, whereas the value of $\sigma_4(p)$ is different.

D. Time Step Calculation

Using central space and forward time differencing, the stability limitation for the model one-dimensional convection equation $u_t + cu_y = 0$ is $(c\Delta t/\Delta y) \leq 1$ [Courant-Friedrichs-Lewy (CFL) limitation], whereas the corresponding stability restriction for the one-dimensional model diffusion equation $u_t = \nu u_{yy}$ is $(\nu\Delta t/\Delta y^2) \leq 1/2$.

In the present scheme, a combination of the two limitations is employed. Specifically,

$$\Delta t = \omega \min\left[\frac{\Delta m}{|u| + (\nu/\alpha\Delta m)}, \frac{\Delta l}{|v| + (\nu/\alpha\Delta l)}\right] \tag{23}$$

where Δm and Δl are the cell dimensions in the m and l local cell directions, u and v are the corresponding velocity components, ν is the kinematic viscosity coefficient, and $\alpha = 1/2$ is the diffusion stability limitation. Lastly, ω is a safety factor and equal to 0.9.

E. Boundary Conditions

Four types of conditions have been applied for the cases considered in the present work. Those are 1) wall, 2) far field, 3) inflow, and 4) outflow. They are applied to the two velocity components, as well as to the pressure corrections.

Specification of pressure, such as in the outflow boundary, is applied through specification of the pressure corrections ϕ using Eq. (8) $p^{(n+1)} - p^{(n)} = (1/\Delta t)\phi$. The ϕ values are simply set to zero. In this way, the pressure values are the initially prescribed ones. Similarly, specification of velocity, such as in the inflow and wall, is applied by using Eq. (10) $\boldsymbol{u}^{(n+1)} = \boldsymbol{u}' - \nabla\phi$ by setting $\nabla\phi = 0$. The velocity values are the initial values.

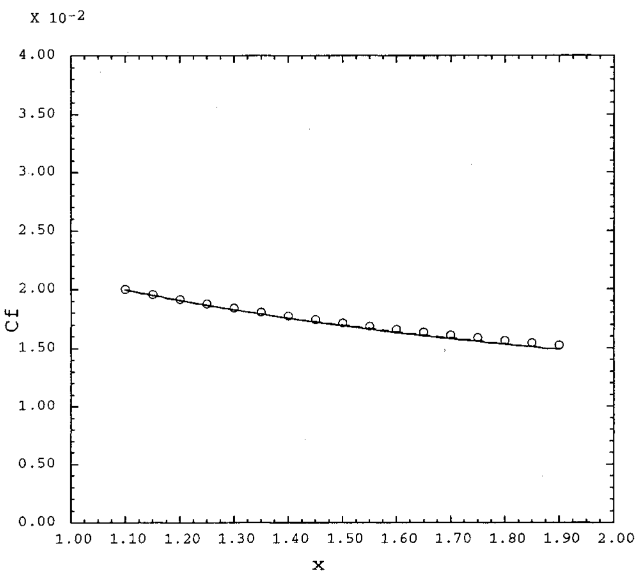

Fig. 3 Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^3$: —, twice globally adapted grid; – –, two-level adapted grid; and o, Blasius.

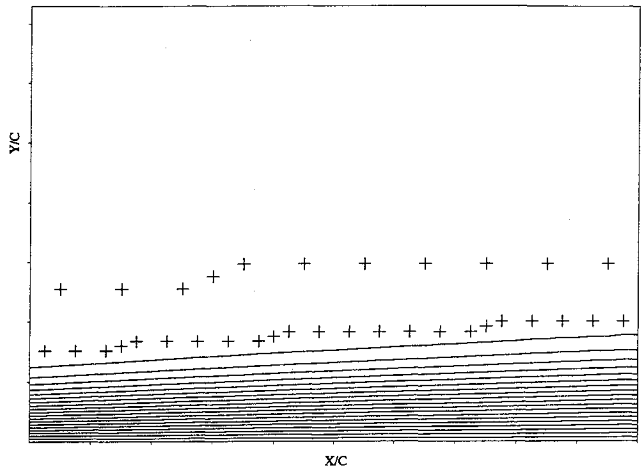

Fig. 4 Velocity contours for Blasius boundary layer with $Re = 10^3$, with a two-level adapted grid (the + symbols indicate interface modes).

At a wall, the u and v components of velocity are set to zero.The value of $\partial\phi/\partial n$ is also set to zero. At a far field, the velocity components are set equal to the freestream values, whereas the pressure corrections are set to zero.

At an inflow boundary, the u and v components of velocity are specified and $\partial\phi/\partial n$ is set to zero. At an outflow boundary, the velocity gradients in the normal direction to the boundary are set to zero. The pressure corrections are set to zero.

IV. Adaptive Hybrid Grids

The objective of adaptive grid refinement is to adjust the grid scale in regions where extra resolution is needed. The algorithm detects important flow features, such as boundary layers and vortices. Then the cells are divided into smaller children cells. Points are inserted in the middle of the quadrilateral edges.

The resulting embedded grids are topologically similar to the initial grid and so maintain its geometric properties (e.g., stretching, orthogonality) but are not necessarily aligned to the initial grid as the embedded meshes follow the features. The process can be repeated any number of times and results in finer and finer local embedded grids until a region is adequately resolved.[14]

The locations of such features are not known a priori, and they have to be detected. A feature detector senses the flow features that are present in different regions and guides the adaptive algorithm to embed these regions if the existing grid spacing in such regions is not sufficient for resolving the local flow variations.

Local embedding implies two important consequences for any basic solver that is used. First, the mesh now becomes unstructured, and the usual i,j indexing can no longer be used. A system is required to keep track of all of the required information for each cell (pointer system). Second, there is an implied communication between the grids. The borders between grids of different refinement levels (interfaces) must receive special attention. Employment of nonstaggered grids simplifies treatment of grid interfaces.

A. Flow Feature Detection

The feature detector uses velocity differences and velocity gradients across the grid cells for sensing the flow features.[17] Threshold values for the refinement are set based on the distribution of the detection parameters over the cells of the domain. The average S_{ave} and the standard deviation S_{sd} are employed to calculate a threshold S_{th} as follows:

$$S_{\text{th}} = S_{\text{ave}} + \alpha S_{\text{sd}} \tag{24}$$

where S is the detection parameter. The average and the standard deviation are defined as

$$S_{\text{ave}} = \frac{1}{N_{\text{cells}}} \sum_{e=1}^{N_{\text{cells}}} |S_e| \tag{25}$$

$$S_{\text{sd}} = \sqrt{\frac{1}{N_{\text{cells}}} \sum_{e=1}^{N_{\text{cells}}} S_e^2} \tag{26}$$

The value of the parameter α is chosen empirically, with a typical value of the parameter being 0.3. The cells that have a detection parameter value greater than the threshold value are flagged to be refined. The value is such that excessive refinement is avoided, whereas all of the essential flow features are captured.[17] Large values of α may result in inadequate refinement of the regions of the flow features, whereas very small values may result in excessive refinement. Details of this method are given in Ref. 17.

B. Treatment of Grid Interfaces

Embedding of quadrilateral cells introduces internal boundaries between cells with different refinement levels. Grid interfaces are characterized by an abrupt change in cell size. Following division of a portion of the grid cells, the resulting grid contains a number of cells that are left with mid-edge nodes on some of their four edges due to refinement of the neighboring cells. These interface cells constitute the border between the divided and the undivided cells. Employment of an accurate and stable numerical treatment of interfaces is one of the major issues of adaptive-grid algorithms.[19,24,25]

Numerical schemes employ normal quadrilateral cells with four corner nodes, and significant changes are necessary for the scheme to be applied to interface cells with additional mid-edge nodes.[19] This is not desired, as then the adaptive algorithm becomes dependent on the specific numerical scheme that is employed.

Another important issue is maintaining conservation across interfaces. The fluxes across the boundaries surrounding an interface cell should cancel one another for the scheme to be conservative.

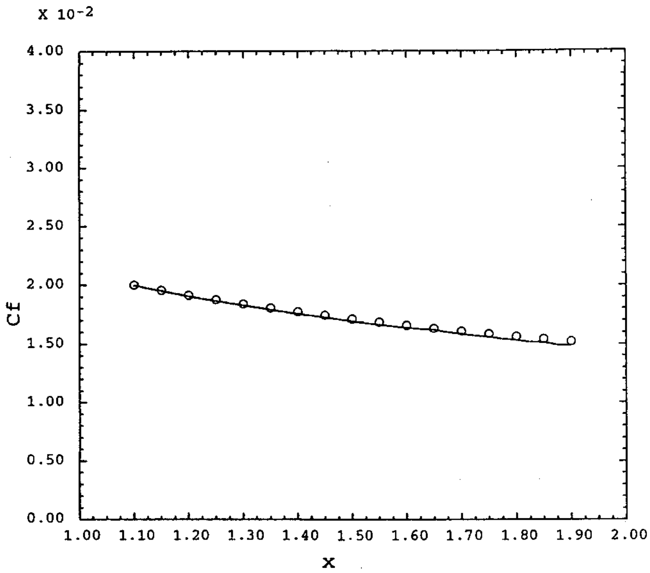

Fig. 7 Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^3$. Interfaces are located inside of the boundary layer; —, one-level adapted grid and o, Blasius.

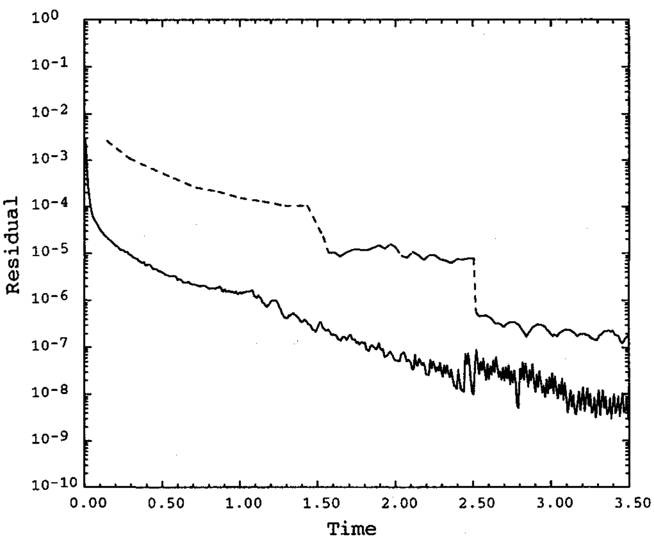

Fig. 5 Maximum residual history for Blasius boundary layer with $Re = 10^3$: —, twice globally adapted grid and – –, two-level adapted grid.

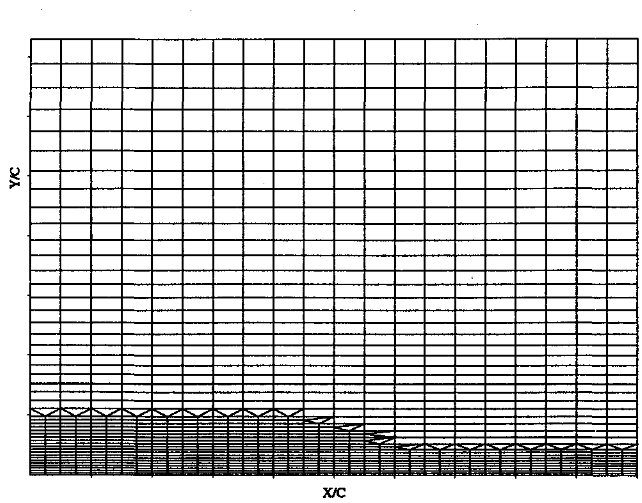

Fig. 6 One-level adapted grid for Blasius boundary layer with $Re = 10^3$. Interfaces are located inside of the boundary layer (initial grid has 21 × 37 points).

C. Hybrid grid

The existence of hanging nodes at the grid interfaces leads to relatively complicated numerical treatments that have accuracy problems.[19] In the present work the approach is to eliminate these hanging nodes and to treat the interface cells using the same integration method as for the rest of the grid cells.

A simple method for eliminating the interface nodes has been developed in the present work. The method divides the interface coarse cells, such as those illustrated in Fig. 1, into smaller triangles. These triangles have as their vertices both the corners of the original interface cell as well as the mid-edge nodes. In this way, the mid-edge points are eliminated, and the grid becomes continuous. No additional points are inserted. Following adaptation of the quadrilaterals, the triangles are created. The triangles corresponding to previous adaptations, which require refinement, are deleted, and corresponding quadrilaterals are refined. The solver is general to handle both types of elements, namely, quadrilaterals and triangles. The mass and momentum fluxes are conserved across the edges of all cells. A linear triangular element is introduced to integrate over the triangle region. Fourth-order difference smoothing is also introduced into both momentum and pressure correction equations for the triangular elements.

V. Validation Test Cases

Two flow geometries are employed to provide an assessment of accuracy and robustness of the developed solver with adaptive hybrid grids: flat plate and cylinders. All of the computations were performed on an IBM RS/6000 workstation. The CPU time is 1.2×10^{-4} s/cell/step.

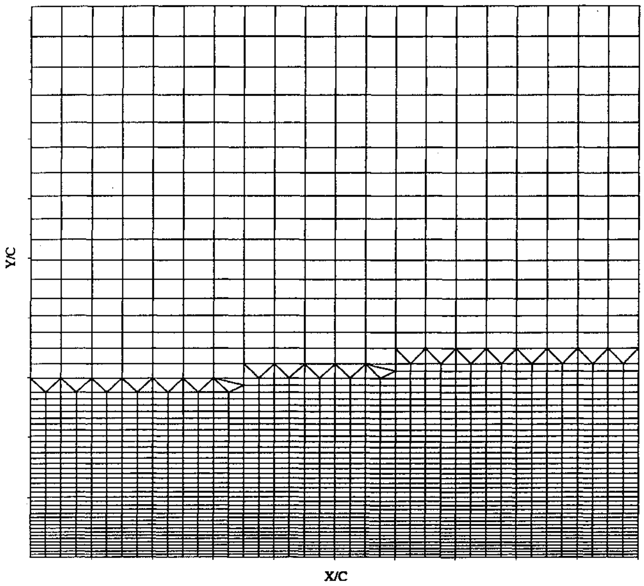

Fig. 8 One-level adapted grid for Blasius boundary layer with $Re = 10^4$; 1960 nodes, 1863 quadrilaterals, and 62 triangles (initial grid has 21 × 37 points).

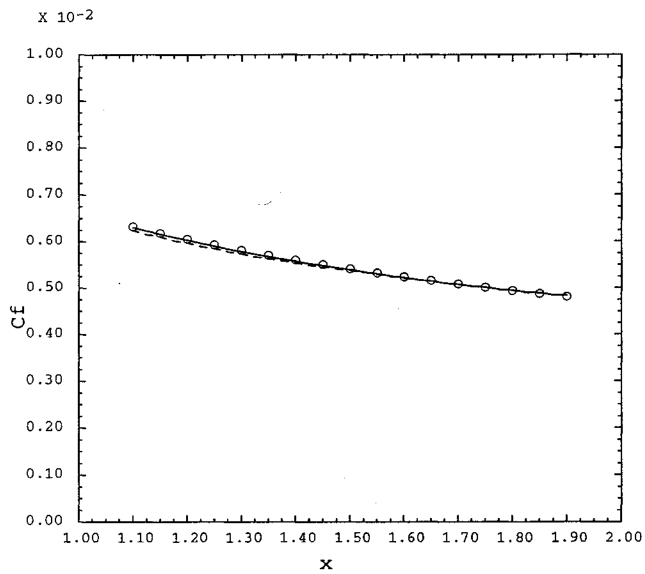

Fig. 9 Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^4$: —, once-globally adapted grid; – –, one-level adapted grid; and o, Blasius.

A typical number of iterations required by the ICCG method for the Poisson equation is 3, whereas 10 iterations are typically required for convergence of the ϕ values. The convergence criterion for the pressure corrections is $\nabla \cdot u < 10^{-5} \cdot U/L$ where U and L are representative velocity and length.

The convergence criterion for the momentum equations is $R_{max} < 10^{-6}$ where R_{max} is the maximum change in time $[u^{(n+1)} - u^{(n)}]$ of u velocity. In steady-state cases, the problems are considered as unsteady, and calculations are carried out until the convergence criterion for the momentum equations is satisfied.

The value of the smoothing coefficients for the momentum and Poisson equations $[\sigma_4(u)$ and $\sigma_4(p)]$ are 10^{-4} and 10^{-3}, respectively. A range of values were tested. The chosen values are the largest values that do not affect accuracy of the solution.

A. Flat Plate Boundary Layer

Blasius boundary layers[26] are computed for a flat plate for different values of the Reynolds number. A Blasius profile is specified at the inlet ($x = 1.0$) to avoid the leading-edge singularity. The computational domain is considered from $x = 1.0$ to 2.0. On the flat plate, wall boundary conditions are applied. At the $x = 1.0$ and $y = y_{max}$ boundaries, outlet boundary conditions are applied, and the pressure p is assumed to be constant due to the Blasius assumption.

Low Re

The first case is a Blasius boundary layer with an inlet Reynolds number of 10^3 ($Re_{x=1.0} = U_\infty \cdot 1/\nu = 10^3$). An initial coarse grid of 11 × 19 points is employed. The smallest grid normal spacing at the wall is 0.016. It is adapted twice as shown in Fig. 2. The grid is hybrid, consisting of both quadrilateral and triangular elements. This grid consists of 1530 nodes, 1443 quadrilaterals, and 124 triangles. The same initial grid of 11 × 19 points is globally refined twice. This grid has the same resolution at the wall as the finest embedded grid of Fig. 2, but it contains no interfaces. This twice globally refined grid has 2993 nodes and 2880 quadrilaterals.

Figure 3 shows computed and analytical skin-friction distributions at the wall of the Blasius solution, the result with the twice globally refined mesh with no interfaces, and the two-level adapted mesh.The agreement is very good. Figure 4 shows the velocity contours with the location of grid interfaces. There are no

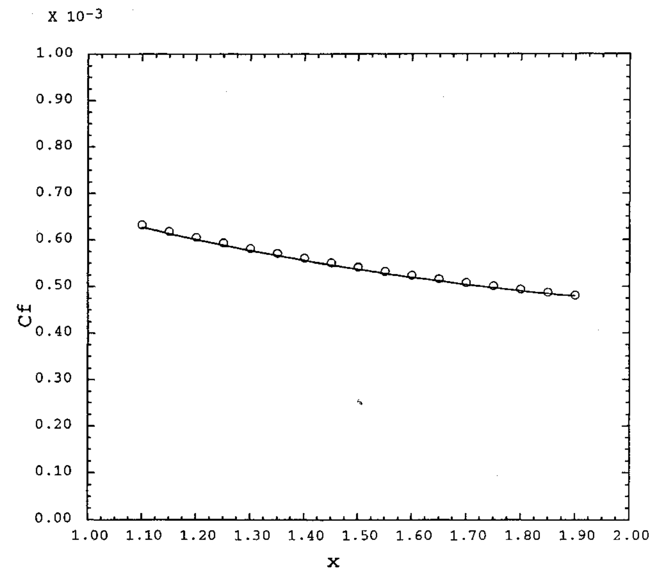

Fig. 10 Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^6$: —, numerical result and o, Blasius.

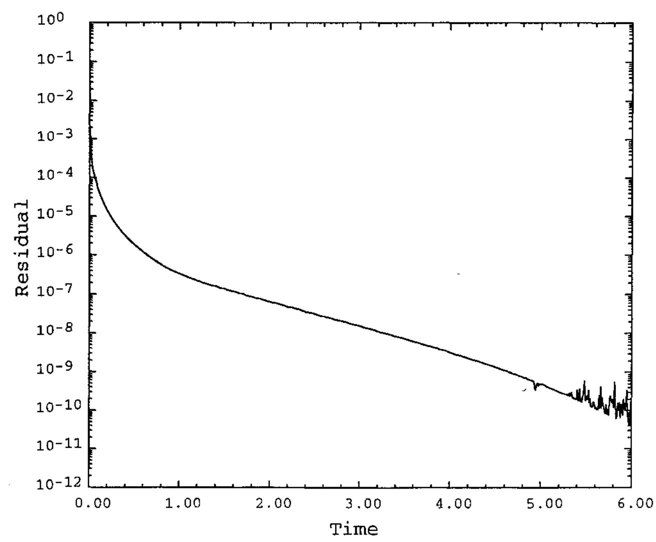

Fig. 11 Maximum residual history for Blasius boundary layer with $Re = 10^6$.

spurious contours observed at the interface regions, where the flow should be uniform. Figure 5 presents the maximum residual histories of the calculation using the twice globally refined grid and the two-level adapted grid. The two convergence histories have similar slopes. The residual corresponding to the two-level adapted grid is higher, since the size of the cells is larger in the initial coarse grid. The initial grid is adapted at time = 1.50 and 2.50.

An initial coarse grid of 21 × 37 points is adapted once as shown in Fig. 6. In this case, interfaces are set inside the boundary layer. Figure 7 shows computed and analytical skin-friction distributions at the wall between the Blasius solution and the result with the one-level adapted mesh. The agreement appears to be very good in both cases.

Medium Re

The second case is a Blasius boundary layer with an inlet Reynolds number of 10^4. An initial grid of 21 × 37 points is employed. The smallest grid normal spacing at the wall is 0.0025. It is

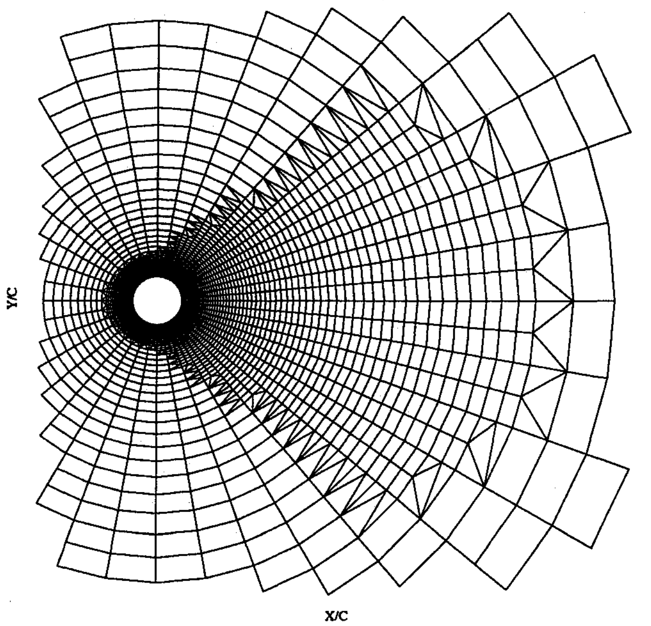

Fig. 12 One-level adapted grid for cylinder flow with *Re* = 40 (initial grid has 35 × 36 points).

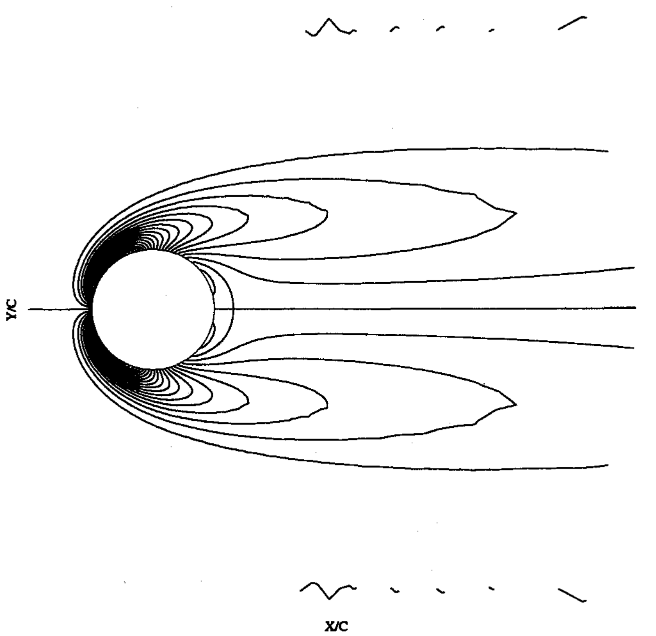

Fig. 12 One-level adapted grid for cylinder flow with *Re* = 40 (initial grid has 35 × 36 points).

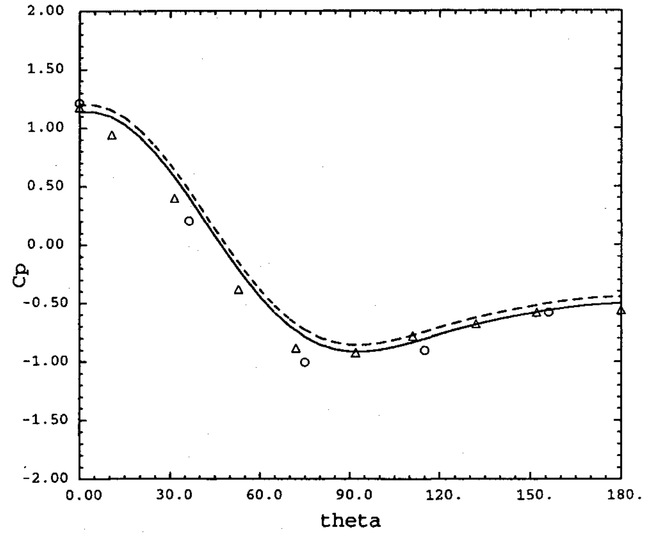

Fig. 14 Comparison of pressure coefficient distributions for cylinder flow with *Re* = 40: —, numerical result with fine mesh; --, adapted mesh; o, experimental result (*Re* = 36); and Δ, experimental result (*Re* = 45).

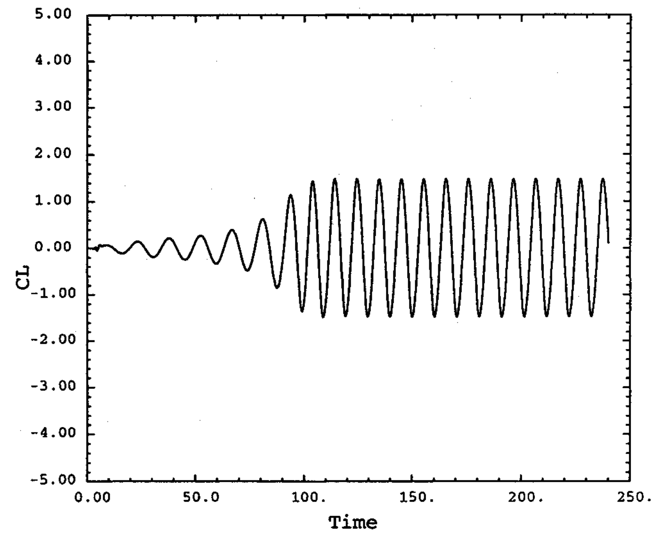

Fig. 15 Crossflow force coefficient (C_L) history —cylinder flow with *Re* = 1.62 × 10^4.

adapted once as shown in Fig. 8. The grid is hybrid, consisting of both quadrilateral and triangular elements. The same initial grid is globally adapted once. Figure 9 shows good comparison of the skin-friction distributions at the wall computed with the once globally adapted grid and the one-level adapted grid.

High Re

The third case is a Blasius boundary layer with an inlet Reynolds number of 10^6. Actually, the flow over a flat plate becomes turbulent if *Re* is larger than 3.0×10^5. The purpose of this laminar case is to test stability of the present scheme with high Reynolds numbers. A fine grid of 41 × 73 points is employed. The smallest grid spacing at the wall is 1.26×10^{-4}.

Figure 10 shows good comparison of computed skin-friction distribution at the wall with that from the analytical Blasius solution. Robustness of the solver is examined in Fig. 11, which illustrates the convergence history. The solution converges to machine accuracy for this high-Reynolds-number case.

B. Cylinder in Crossflow

Flow around circular cylinders is of significant engineering interest. Steady as well as unsteady laminar flow is computed for

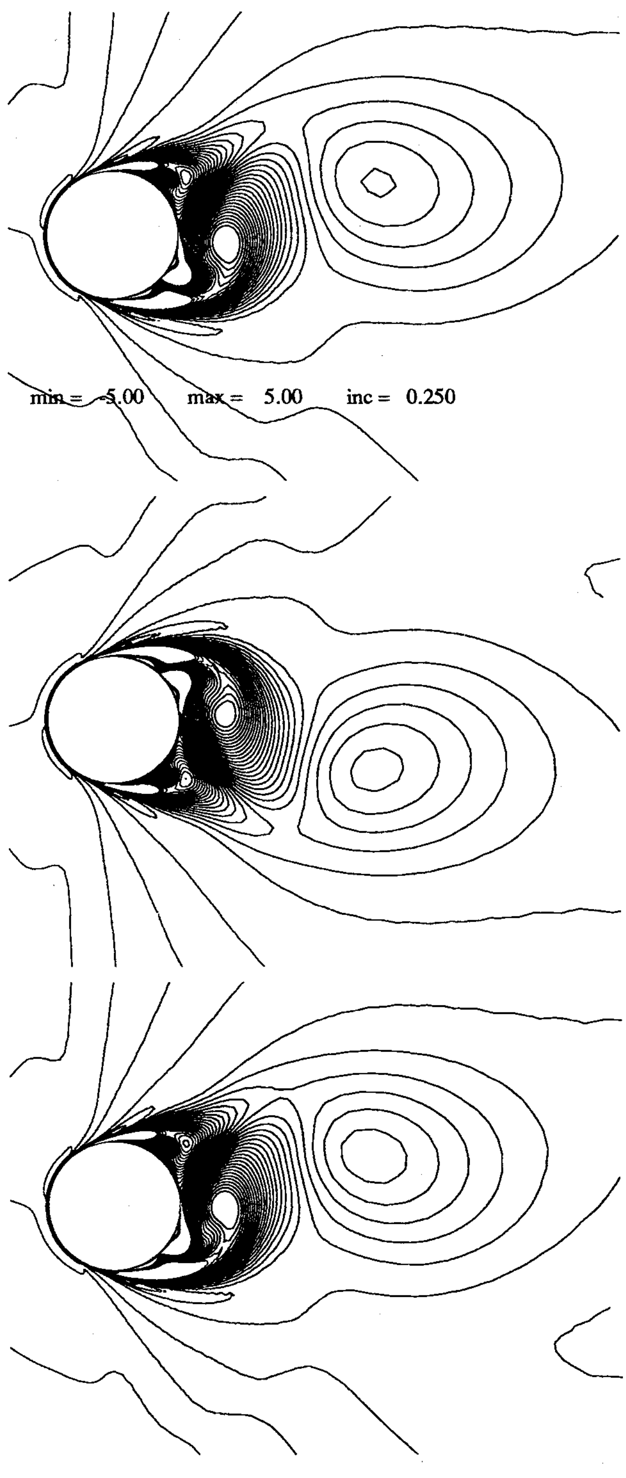

Fig. 16 Stages of vortex shedding (vorticity contours)—cylinder flow with $Re = 1.62 \times 10^4$ (t = 230, 235, and 240).

different Reynolds numbers. The boundary layer on the surface separates, forming vortices. These vortices are stationary for low Reynolds numbers (up to $Re = 40$) and are convected downstream, forming a characteristic wake for higher Reynolds numbers. The pressure and velocity fields are computed and compared with experimental results. Furthermore, the unsteady forces on the cylinder for the cases of vortex shedding are monitored. The frequency of the regular oscillation f, when nondimensionalized by the diameter D of the cylinder and the freestream velocity U, is called the Strouhal number:

$$St = fD/U_\infty \tag{27}$$

The Strouhal number is roughly 0.20 over a wide range of Reynolds numbers.[27] This Strouhal number of the shedding is compared with experiments.

Steady Flow of Re = 40

The first case is a steady cylinder flow with a Reynolds number of 40. A fine grid of 52 × 72 points is employed. The smallest grid normal spacing at the wall is 0.04, and a constant grid stretching factor of 1.10 is employed in the normal to the surface direction. The far-field boundary is placed 15 diameters away from the cylinder.

An initial coarse grid of 35 × 36 points is adapted once as shown in Fig. 12. The grid is hybrid, consisting of both quadrilateral and triangular elements. The flowfield features two large vortices located symmetrically downstream of the cylinder (Fig. 13).

The calculated C_D (drag coefficient) is 1.50, and this is within the range of the experimental results presented in Ref. 28, which range between 1.40 and 1.70. Figure 14 shows good comparison of computed pressure coefficient distribution on the cylinder between the numerical results for the fine and adapted grids and the experimental results.[29]

Unsteady Flow of Re = 1.6×10^4

A fine grid of 111 × 144 points is employed. The smallest grid normal spacing at the wall is 2.00×10^{-3}, and a constant grid stretching factor of 1.10 is employed in the normal to the surface direction. The far-field boundary is placed 15 diam away from the cylinder.

Figure 15 shows the history of the C_L (crossflow force coefficient). The Strouhal number is 0.21 in this case, which coincides with the experimental value.[27] The value of f in Eq. (27) is obtained from the average period of the C_L history. Figure 16 shows the different stages of vortex formation and shedding. Vortices are formed alternatively on both the upper and lower parts of the cylinder surface. Then they are convected downstream, while interacting with each other.

VI. Concluding Remarks

The developed finite element, explicit/implicit marching scheme for the unsteady two-dimensional Navier-Stokes equations of incompressible fluid flow yielded stable and accurate results for the test cases considered. The nonstaggered grid that was employed made the method suitable for an adaptive algorithm, which employed locally embedded meshes.

A combination of quadrilateral as well as triangular cells provided flexibility in forming the adaptive grids. The numerical treatment of grid interfaces with employment of triangular elements proved to be stable, even in cases in which the interfaces were located within the boundary-layer regions. Comparisons with equivalent globally fine grids that contain no interfaces provided an evaluation of accuracy.

Applications of the developed adaptive algorithm included both steady and unsteady flows. Comparisons with analytical and experimental results provided further evaluation of accuracy and robustness of the developed adaptive method.

Acknowledgments

This work was supported by the National Science Foundation (NSF) Engineering Research Centers Program Grant CDR-8721512 as well as by NSF Grant ECS-9023770.

References

[1]Gresho, P. M., "Some Current CFD Issues Relevant to the Incompressible Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering,* Vol. 87, 1991, pp. 201–252.

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

[3]Choi, D., and Merkle, C. L., "Application of Time-Iterative Schemes to

Incompressible Flow," *AIAA Journal,* Vol. 23, No. 10, 1985, pp. 1518–1524.

[4]Kwak, D., and Chakravarthy, S. R., "A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables," *AIAA Journal,* Vol. 24, No. 3, 1986, pp. 390–396.

[5]Harlow, F. H., and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface," *Physics of Fluids,* Vol. 8, No. 12, 1965, pp. 2182–2189.

[6]Patankar, S., and Spalding, D., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows," *International Journal of Heat and Mass Transfer,* Vol. 15, No. 3, 1972, pp. 1787–1806.

[7]Raithby, G. D., and Schneider, G. E., "Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling," *Numerical Heat Transfer,* Vol. 2, No. 2, 1979, pp. 417–440.

[8]Kim, S. W., and Benson, T. J., "Comparison of the SMAC, PISO and Iterative-Advancing Schemes for Unsteady Flows," *Computers & Fluids,* Vol. 21, No. 3, 1992, pp. 435–454.

[9]Patankar, S. V., *Numerical Heat Transfer and Fluid Flow,* Hemisphere, New York, 1980.

[10]Fletcher, C. A. G., *Computational Galerkin Method,* Springer-Verlag, New York, 1984.

[11]Perić, R. M., Kessler, G., and Scheuerer, G., "Comparison of Finite-Volume Numerical Methods with Staggered and Collocated Grids," *Computers & Fluids,* Vol. 16, No. 4, 1988, pp. 389–403.

[12]Hafez, M., and Soliman, M. "Numerical Solution of the Incompressible Navier-Stokes Equations in Primitive Variables on Unstaggered Grids," AIAA Paper 91-1561, July 1991.

[13]Franca, L. P, and Frey, S. L., and Hughes, T. J. R., "Stabilized Finite Element Methods: II. The Incompressible Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering,* Vol. 99, No. 2, 1992, pp. 209–233.

[14]Zienkiewicz, O. C., and Taylor, R. L., *The Finite-Element Method,* 4th ed. Vol. 2, McGraw-Hill, New York, 1991, Chap. 13.

[15]Dannenhoffer, III, J. F., and Baron, J. R., "Grid Adaptation for the 2-D Euler Equations," AIAA Paper 85-0484, Jan. 1985.

[16]Löhner, R., "The Efficient Simulation of Strongly Unsteady Flows by the Finite-Element Method," AIAA Paper 87-0555, Jan. 1987.

[17]Kallinderis, Y., and Baron, J. R., "Adaptation Methods for a New Navier-Stokes Algorithm," *AIAA Journal,* Vol. 27, No. 1, 1989, pp. 37–43.

[18]Kallinderis, Y., "Algebraic Turbulence Modeling for Adaptive Unstructured Grids," *AIAA Journal,* Vol. 30, No. 3, 1992, pp. 631–639.

[19]Kallinderis, Y., "Numerical Treatment of Grid Interfaces for Viscous Flows," *Journal of Computational Physics,* Vol. 98, No. 1, 1992, pp. 129-144.

[20]Mavriplis, D. J., "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes," *AIAA Journal,* Vol. 28, No. 2, 1990, pp. 213–221.

[21]Oguni, T., Murata, T., Miyoshi, T., Dongarra, J. J., and Hasegawa, H., *Matrix Computation Software,* Maruzen, 1991.

[22]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite-Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, Jan. 1981.

[23]Kallinderis, Y., "A Finite Volume Navier-Stokes Algorithm for Adaptive Grids," *International Journal for Numerical Methods in Fluids,* Vol. 15, No. 2, 1992, pp. 193–217.

[24]Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Zonal Flow Calculations," AIAA Paper 84-0164, Jan. 1984.

[25]Allmaras, S. R., and Baron, J. R., "Embedded Mesh Solutions of the 2-D Euler Equations: Evaluation of Interface Formulations," AIAA Paper 86-0509, Jan. 1986.

[26]Schlichting, H., *Boundary-Layer Theory,* 7th ed., McGraw-Hill, New York, 1979.

[27]Sarpkaya, T., and Issacson, M., *Mechanics of Wave Forces on Offshore Structures,* Van Nostrand Reinhold, Amsterdam, 1981.

[28]Braza, P., Chassaing, P., and Ha Minh, H., "Numerical Study and Physical Analysis of the Pressure and Velocity Fields in the Near Wake of a Circular Cylinder," *Journal of Fluid Mechanics,* Vol. 165, No. 1, 1986, pp. 79–130.

[29]Grove, A. S., Shair, F. H., and Petersen, E. E., "An Experimental Investigation of the Steady Separated Flow Past a Circular Cylinder," *Journal of Fluid Mechanics,* Vol. 19, Pt. 1, 1964, pp. 60–80.