Computer Assisted Mechanics and Engineering Sciences, 1: 123–146, 1994.

A finite element method for the incompressible Navier-Stokes equations with adaptive hybrid grids

Kengo Nakajima
Research Center for Safety Science, Mitsubishi Research Institute, Inc.
1-8-1 Shimo-Meguro, Meguro-ku, Tokyo 153, Japan

Yannis Kallinderis
Dept. Aerospace Engineering and Engineering Mechanics,
The University of Texas at Austin, Austin, Texas 78712, USA

(Received August 1, 1994)

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 non-staggered grid. The solution is advanced in time with an explicit/implicit marching scheme. An adaptive algorithm has been implemented, which refines the grid locally in order 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.

1. 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 [8]. 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, while 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 3-D, 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 due to the fact that the speed of sound approaches infinity at the incompressible limit. A method that uses compressible-like governing equations is the artificial compressibility approach [3, 4, 18]. A time derivative of the pressure is added to the continuity equation and the incompressible flow field 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 [11, 17, 22, 26]. 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. In order to reject these modes, *staggered grids* have been employed by several of these algorithms [6, 23]. On the other hand, employment of non-staggered grids [7, 10, 24] 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 [8]. Furthermore extensive literature on finite-element methods is given in [30].

Resolution of the computational mesh plays a crucial role on accuracy of computations. However, generation of a grid which 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, in order 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 to the size of the overall computational domain. Local grid embedding consists of division of cells in order 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 [5, 19], as well as for turbulent compressible flows [13, 14].

One of the most serious problems with using quadrilaterals has been the presence of grid interfaces, which require special numerical treatment [15], which can be quite complicated. Triangular meshes have also been employed with grid embedding for compressible flows [19, 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 in order 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 non-staggered 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 in order to resolve detected flow features. Employment of non-staggered grid facilitates application of adaptive gridding. Applications of the developed adaptive algorithm 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.

2. Governing Equations and Pressure Correction Formulation

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

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

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

An explicit/implicit marching scheme is adopted for integration in time of the above equations. The velocity values are treated explicitly, while the pressure values are treated implicitly in the momentum equations [8]. The velocity values are marched in time with a forward Euler scheme [11]. The continuity equation is formulated implicitly with the velocity values considered at time level $(n+1)$. Specifically, the corresponding semi-discrete system is written as follows:

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

$$\frac{\vec{u}^{(n+1)} - \vec{u}^{(n)}}{\Delta t} + \nabla \cdot \left(\vec{u}^{(n)} \vec{u}^{(n)} \right) = -\nabla p^{(n+1)} + \frac{1}{Re} \Delta \vec{u}^{(n)} \tag{4}$$

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

$$\frac{\vec{u}' - \vec{u}^{(n)}}{\Delta t} + \nabla \cdot \left(\vec{u}^{(n)} \vec{u}^{(n)} \right) = -\nabla p^{(n)} + \frac{1}{Re} \Delta \vec{u}^{(n)} \tag{5}$$

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

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

Introducing a scalar potential ϕ, such that

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

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 equation (7) and considering the continuity equation (3), the following pressure correction Poisson equation is obtained:

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

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

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

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

The above solution procedure follows the explicit/implicit marching scheme in [17]. The overall solution procedure corresponding to marching by one-time-step is summarized as follows:

1. calculate the auxiliary velocity vector $\vec{u}'$ from (5) using $\vec{u}^{(n)}$ and $p^{(n)}$ values
2. solve (9) and obtain the ϕ values
3. calculate $\vec{u}^{(n+1)}$ and $p^{(n+1)}$ using (10) and (11)
4. if $\nabla \cdot \vec{u}^{(n+1)} < \epsilon$ where ϵ is the tolerance for divergence, advance to next time step; if not, consider $\vec{u}^{(n+1)}$ as $\vec{u}'$ and repeat steps 2 and 3.

3. FINITE-ELEMENT DISCRETIZATION

Both momentum and pressure correction equations are discretized using the Galerkin finite-element approach on a *non-staggered* grid. The scheme is compact with all operations being restricted to within each grid-cell. *Linear* triangles and *bilinear isoparametric* quadrilateral elements are employed [6].

3.1. Momentum equations

The momentum equation (5) can be written as

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

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 i-th node. In the present method, u, v and p are defined at cell vertices.

Substituting (13) into (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 [6]:

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

where

$$\begin{aligned} M_{ij} &= \iint_\Omega N_i N_j \, \mathrm{d}x \, \mathrm{d}y \\ K^x_{ij} &= \iint_\Omega N_i \frac{\partial N_j}{\partial x} \, \mathrm{d}x \, \mathrm{d}y \\ K^y_{ij} &= \iint_\Omega N_i \frac{\partial N_j}{\partial y} \, \mathrm{d}x \, \mathrm{d}y \\ D_{ij} &= \iint_\Omega \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) \mathrm{d}x \, \mathrm{d}y \end{aligned}$$

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

$$\begin{aligned} \mathbf{M}\mathbf{u}' &= \mathbf{M}\mathbf{u}^{(n)} - \Delta t\, \mathbf{F}_u \\ \mathbf{M}\mathbf{v}' &= \mathbf{M}\mathbf{v}^{(n)} - \Delta t\, \mathbf{F}_v \end{aligned} \tag{15}$$

where

$$\mathbf{M} = \sum_e M_{ij}$$

$$\mathbf{u}^{\mathrm{T}} = [u_1, u_2, u_3 \cdots], \qquad \mathbf{v}^{\mathrm{T}} = [v_1, v_2, v_3 \cdots]$$

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

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

where the summation is over all the elements. The consistent mass matrix $\mathbf{M}$ is used. Therefore, $\mathbf{u}'$ and $\mathbf{v}'$ are obtained as follows:

$$\begin{aligned} \mathbf{u}' &= \mathbf{u}^{(n)} - \Delta t \, \mathbf{M}^{-1} \mathbf{F}_u \\ \mathbf{v}' &= \mathbf{v}^{(n)} - \Delta t \, \mathbf{M}^{-1} \mathbf{F}_v \end{aligned} \tag{16}$$

3.2. 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 expressions as in (13). In the present method, u, v and ϕ are defined at cell nodes. Integrating equation (9) over the each element domain Ω using the Galerkin method, the following equation is obtained:

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

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 (18)

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

where

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

$$\mathbf{f} = \sum_e \left[K_{ij}^x \, u'_j + K_{ij}^y \, v'_j \right]$$

The values for ϕ on the left hand side of (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 ICCG (*Incomplete Cholesky Conjugate Gradient*) iterative method [21]. The rate of convergence of the iterative process depends on the numbering of nodes. In order to accelerate convergence, the elements of the matrix are renumbered using the RCM (*Reverse Cuthill-Mckee*) method [21]. Finally, the velocity and pressure values are obtained by integrating equations (10) and (11).

3.3. Artificial dissipation

For high Reynolds number flow, the nonlinear convection terms are dominant. Central space differencing schemes are susceptible to oscillatory modes. Furthermore, odd-even decoupling of the solution may appear in the pressure field for this non-staggered type of mesh that is employed.

In the present work, a fourth order smoothing term is added explicitly to the momentum equations in order to suppress odd-even decoupling of the velocity solution [12]. Furthermore, fourth order dissipation is added to the pressure correction equation in order 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:

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

It should be noted that similar cell-based operations of equations (20) and (21) are applied to triangular meshes, as well. The values of u' and v' in momentum equations are updated as follows:

$$\begin{aligned} u_i' &= u_i^{(n)} + \Delta u_i \\ v_i' &= v_i^{(n)} + \Delta v_i \end{aligned} \tag{22}$$

The fourth-order difference terms of u and v are added to stabilize the solution as follows:

$$\begin{aligned} u_i' &= u_i^{(n)} + \Delta u_i + \sigma_4(u) D_i^4\left(u_i^{(n)}\right) \\ v_i' &= v_i^{(n)} + \Delta v_i + \sigma_4(v) D_i^4\left(v_i^{(n)}\right) \end{aligned} \tag{23}$$

where σ_4 is an empirical coefficient. 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 (σ_4). The determined values are such that the solution accuracy is not affected, while the *odd-even* modes are suppressed [16].

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

$$\mathbf{D\Phi} = \mathbf{f} + \mathbf{d} \tag{24}$$

where

$$\mathbf{d}^{\mathrm{T}} = \left[\sigma_4(\mathrm{p})\,\mathrm{D}_\mathrm{j}^4\left(\mathrm{p}_\mathrm{j}^{(\mathrm{n})}\right)\right]$$

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, while the value of $\sigma_4(p)$ is different.

3.4. Time-step calculation

Using central space and forward time differencing, the stability limitation for the model 1-D convection equation $u_t + cu_y = 0$ is $\frac{c\Delta t}{\Delta y} \leq 1$ (CFL limitation), while the corresponding stability restriction for the 1-D model diffusion equation $u_t = \nu u_{yy}$ is $\frac{\nu \Delta t}{\Delta y^2} \leq \frac{1}{2}$.

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

$$\Delta t = \omega \min \left\{ \frac{\Delta m}{|u| + \frac{\nu}{\alpha \Delta m}}, \frac{\Delta l}{|v| + \frac{\nu}{\alpha \Delta l}} \right\}, \tag{25}$$

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

3.5. Boundary conditions

Four types of conditions have been applied for the cases considered in the present work. Those are (*i*) *wall*, (*ii*) *far field*, (*iii*) *inflow* and (*iv*) *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 equation (8) $p^{(n+1)} - p^{(n)} = \frac{1}{\Delta t}\phi$. 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 equation (10) $\vec{u}^{(n+1)} = \vec{u}' - \nabla\phi$ by setting $\nabla\phi = 0$. The velocity values are the initial values.

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 *farfield*, the velocity components are set equal to the free-stream values, while 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.

4. 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 [30].

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 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 *non-staggered* grids simplifies treatment of grid interfaces.

4.1. Flow feature detection

The feature detector uses velocity differences and velocity gradients across the grid-cells for sensing the flow *features* [13]. 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:

$$S_{\text{th}} = S_{\text{ave}} + \alpha S_{\text{sd}} \tag{26}$$

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

$$S_{\text{ave}} = \frac{1}{N_{\text{cells}}} \sum_{e=1}^{N_{\text{cells}}} |S_e| \tag{27}$$

$$S_{\text{sd}} = \sqrt{\frac{1}{N_{\text{cells}}} \sum_{e=1}^{N_{\text{cells}}} {S_e}^2} \tag{28}$$

The value of the parameter α is chosen empirically, with 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, while all the essential flow features are captured [13]. Large values of α may result in inadequate refinement of the regions of the flow features, while very small values may result in excessive refinement. Details of this method are given in [13].

4.2. 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 [1, 15, 25].

Numerical schemes employ normal quadrilateral cells with four corner nodes and significant changes are necessary in order for the scheme to be applied to interface cells with additional mid-edge nodes [15]. 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 in order for the scheme to be conservative.

Hybrid grid

Existence of hanging nodes at the grid interfaces leads to relatively complicated numerical treatments which have accuracy problems [15]. 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

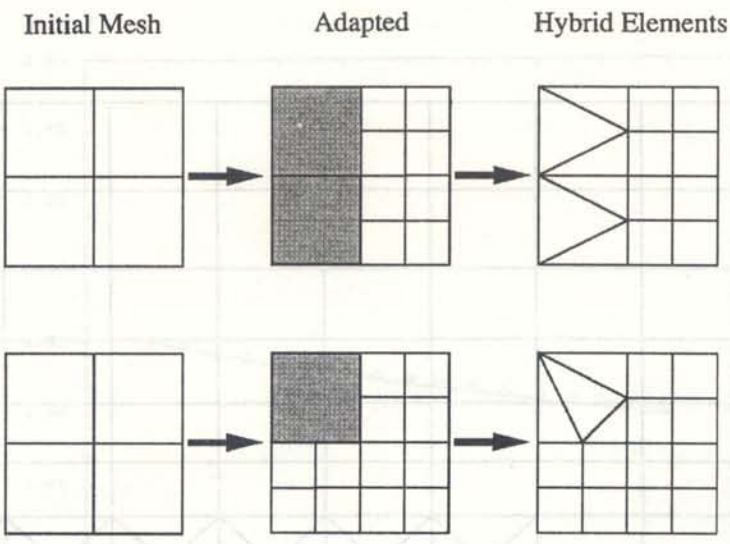


Fig. 1. Elimination of interface nodes by employing triangular elements

of elements, namely quadrilaterals and triangles. The mass and momentum fluxes are conserved across the edges of all cells. 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.

5. Validation test cases

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

A typical number of iterations required by the ICCG method for the Poisson equation is 3, while 10 iterations are typically required for convergence of the ϕ values. Convergence criteria for the pressure corrections is $\nabla \cdot \vec{u} < 10^{-5} \cdot U/L$ 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.

5.1. Flat plate boundary layer

Blasius boundary layers [28] are computed for a flat plate for different values of the Reynolds number. A Blasius profile is specified at the inlet (x=1.0) in order to avoid the leading edge singularity. The computational domain is considered from $x = 1.0$ to $x = 2.0$. 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 ($Re_{x=1.} = U_\infty \cdot 1/\nu = 10^3$). 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

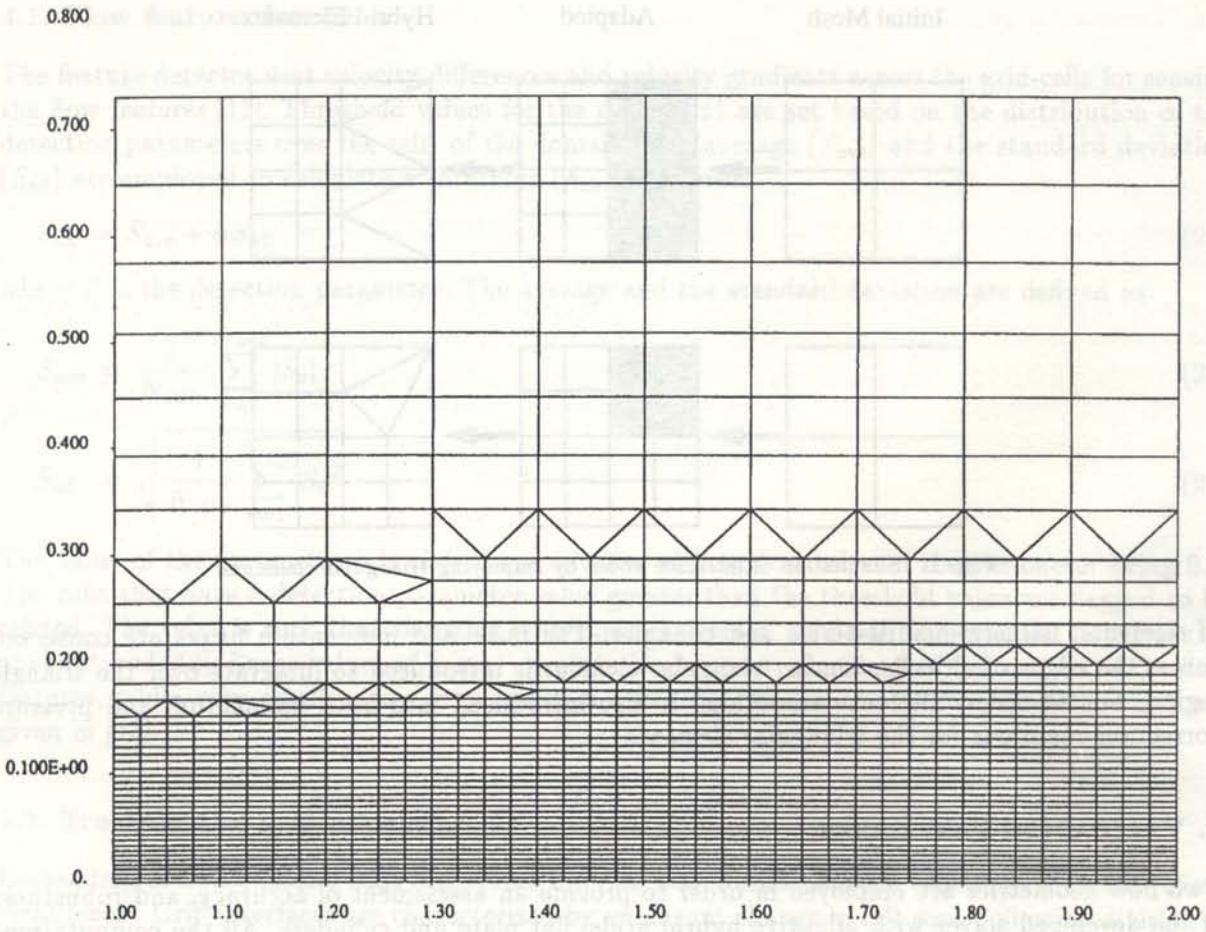


Fig. 2. Two-level adapted grid for Blasius boundary layer with $Re = 10^3$, 1530 points (initial grid has 11×19 points).

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 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 *time* = 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 adapted once as shown in Fig. 8. The grid is hybrid consisting of both quadrilateral and triangular elements. The

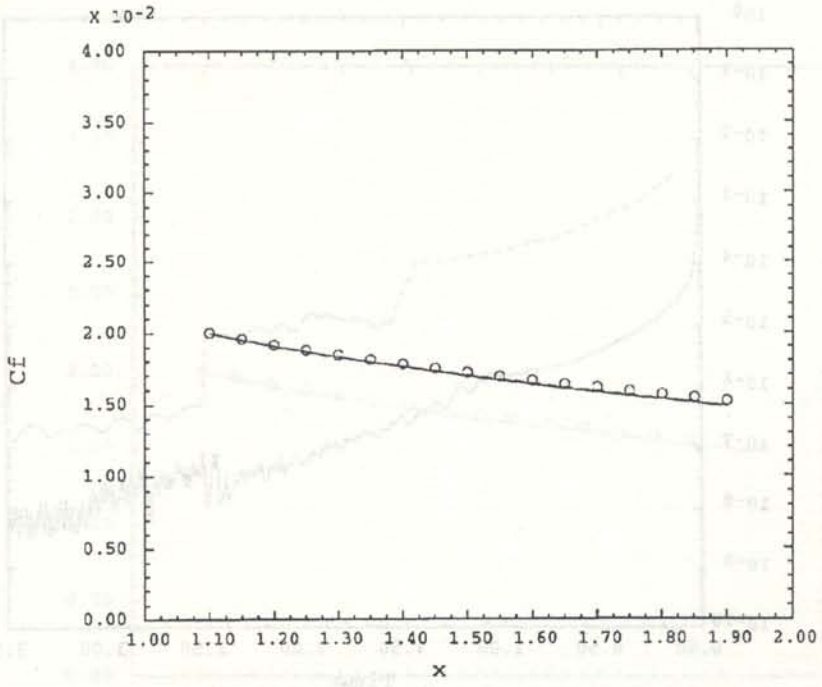


Fig. 3. Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^3$; — twice globally adapted grid, – – 2-level adapted grid, o Blasius

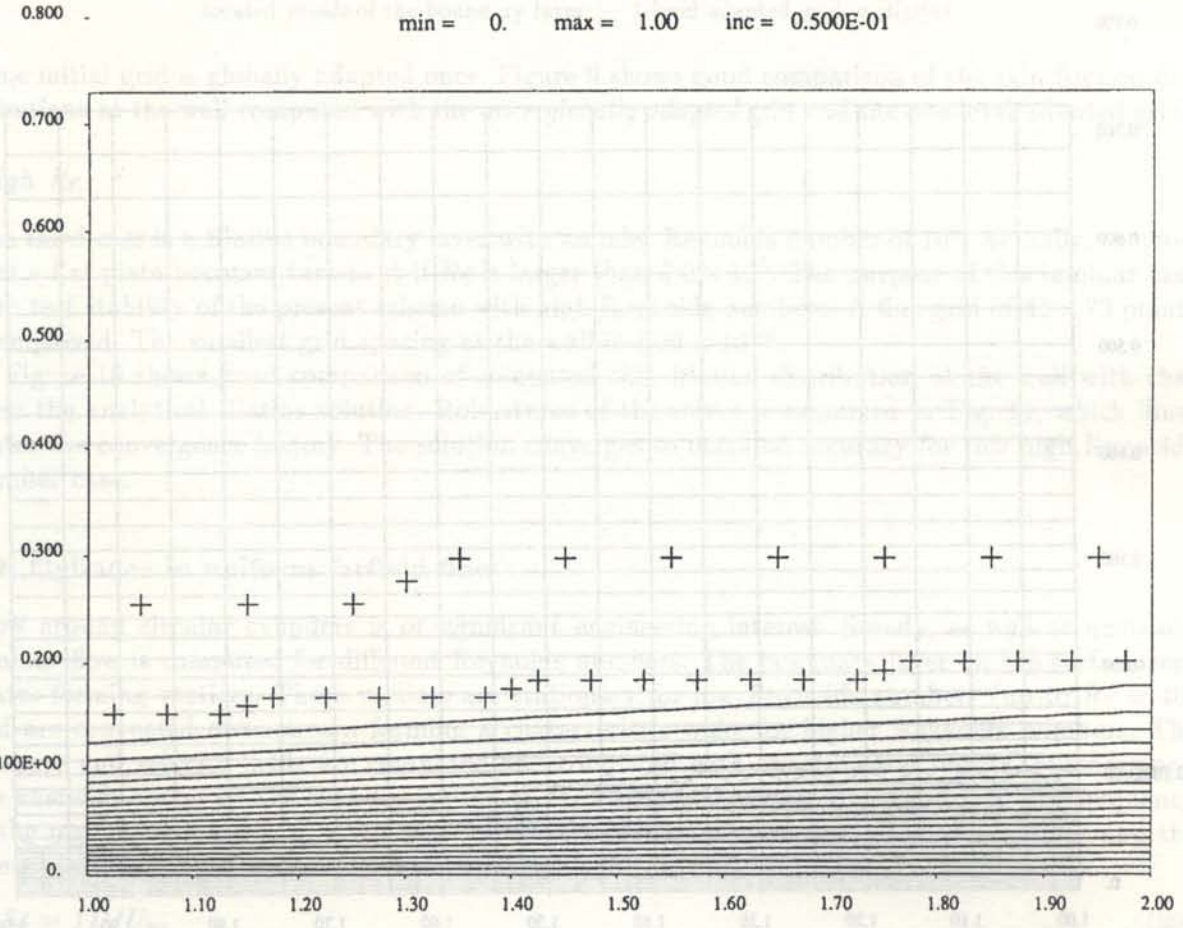


Fig. 4. Velocity contours for Blasius boundary layer with $Re = 10^3$. Numerical result with 2-level adapted grid; + interface nodes

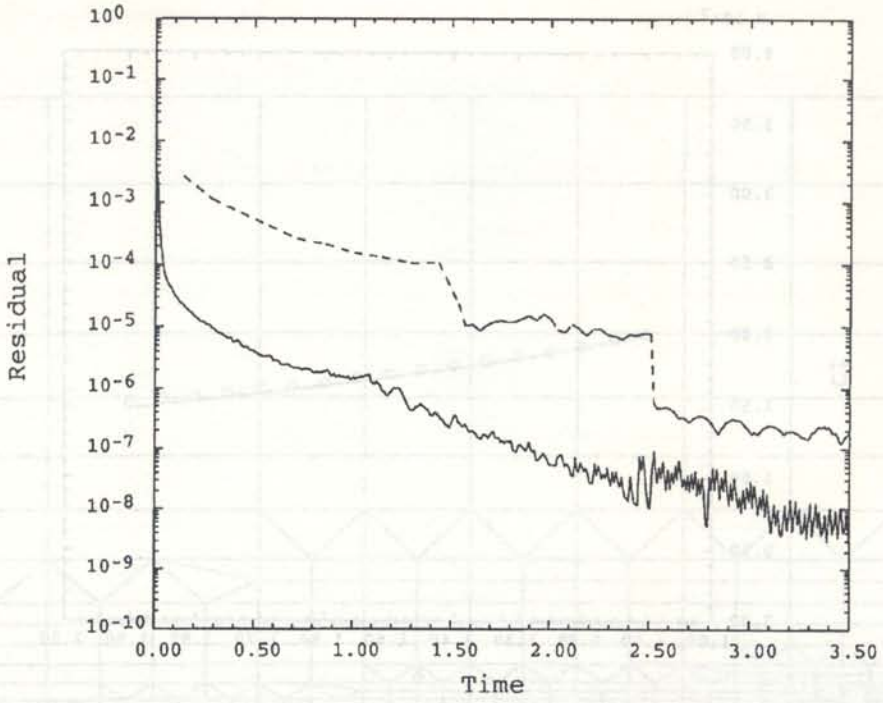


Fig. 5. Maximum residual history for Blasius boundary layer with $Re = 10^3$; — Twice globally adapted grid, – – 2-level adapted grid

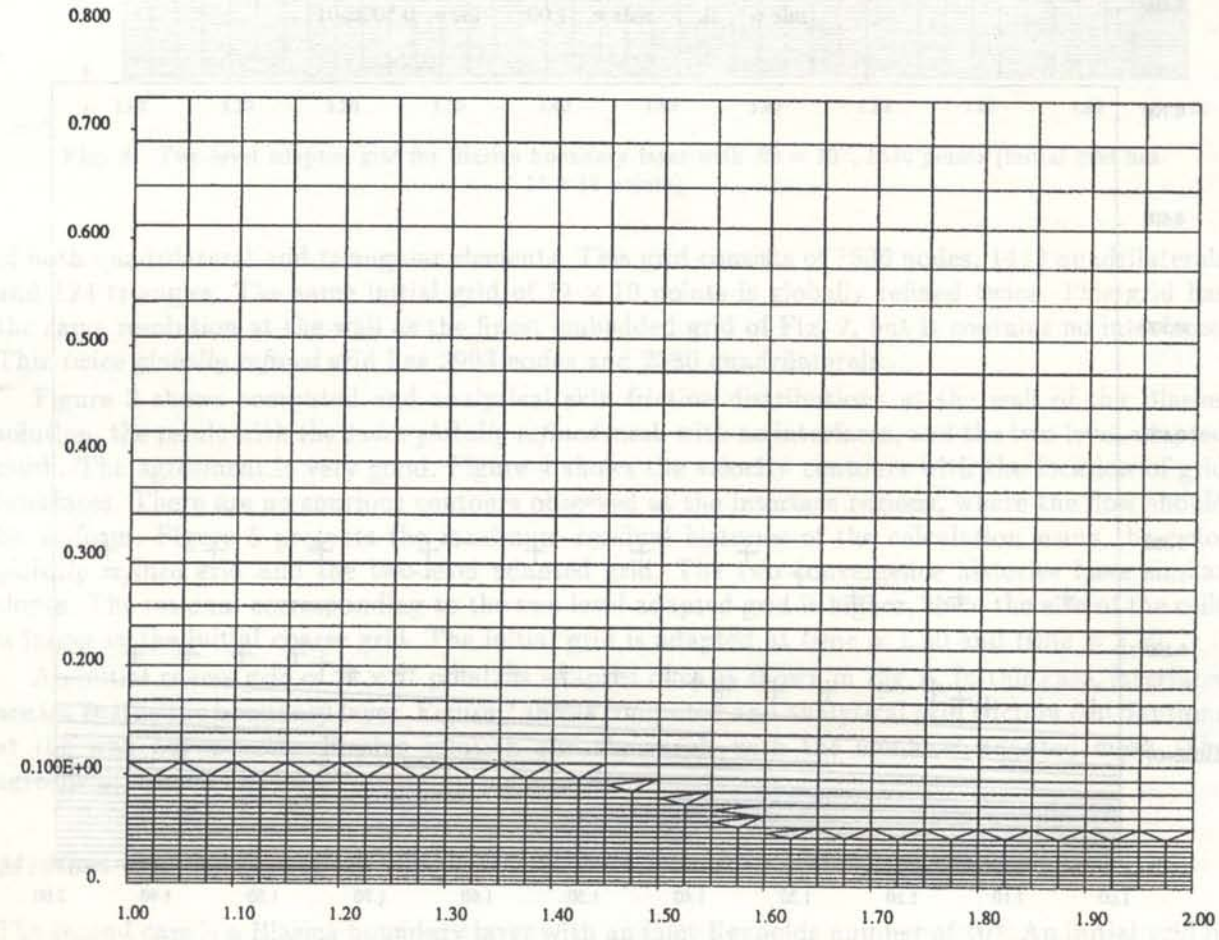


Fig. 6. One-level adapted grid for Blasius boundary layer with $Re = 10^3$. Interfaces are located *inside* of the boundary layer (initial grid has 21×37 points)

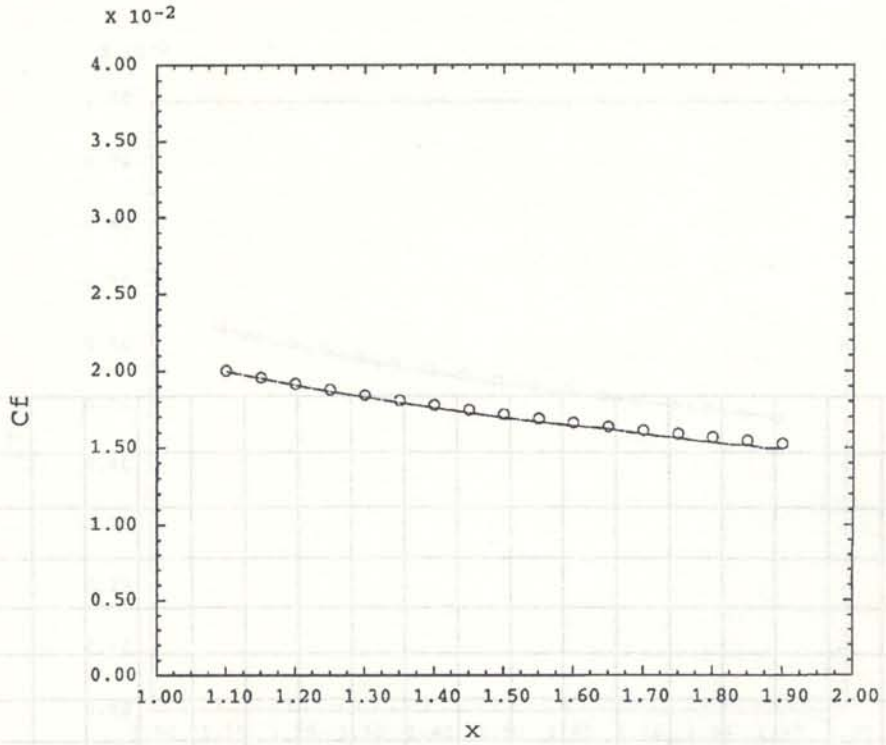


Fig. 7. Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^3$. Interfaces are located *inside* of the boundary layer; — 1-level adapted grid, o Blasius

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.

5.2. Cylinder in uniform farfield flow

Flow around circular cylinders is of significant engineering interest. Steady, as well as unsteady laminar flow is computed for 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 non-dimensionalized by the diameter D of the cylinder and the free-stream velocity U, is called the *Strouhal* number.

$$St = fD/U_\infty \tag{29}$$

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.

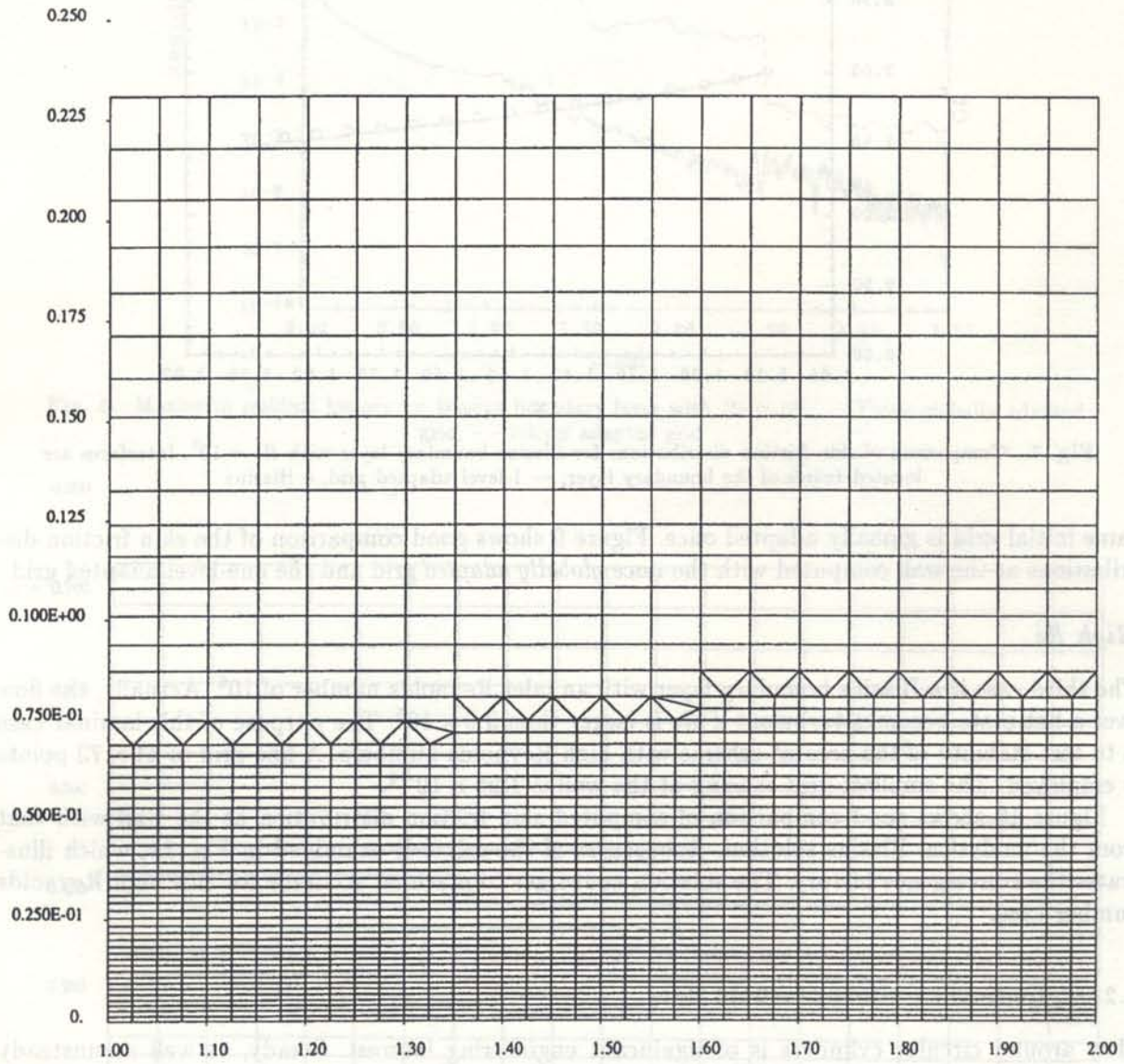


Fig. 8. One-level adapted grid for Blasius boundary layer with $Re = 10^4$, 1960 nodes, 1863 quadrilaterals, 62 triangles (initial grid has 21 × 37 points)

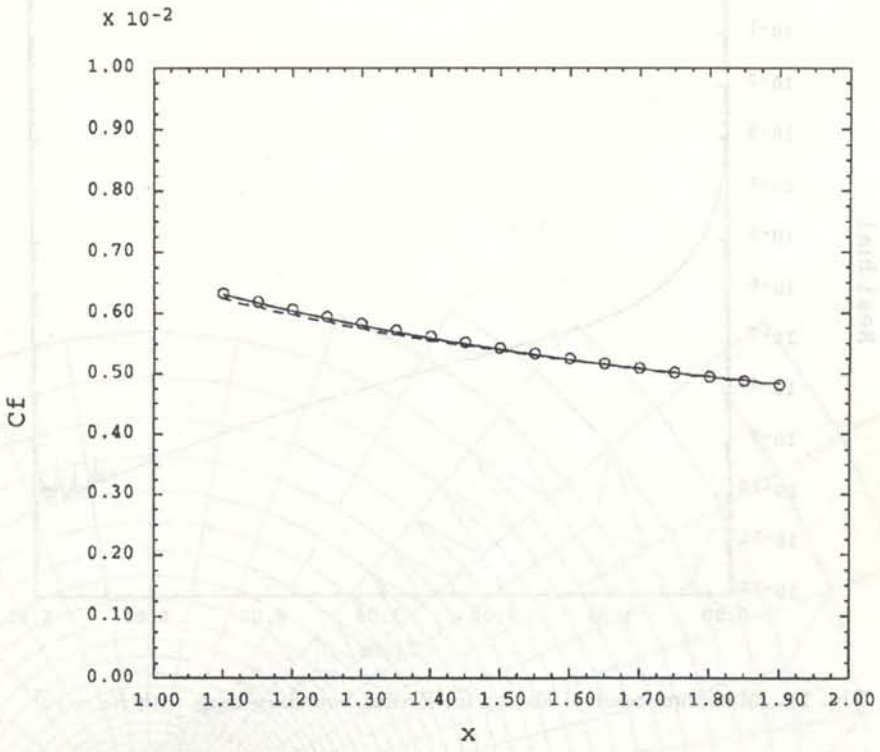


Fig. 9. Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^4$; — once globally adapted grid, – – 1-level adapted grid, o Blasius

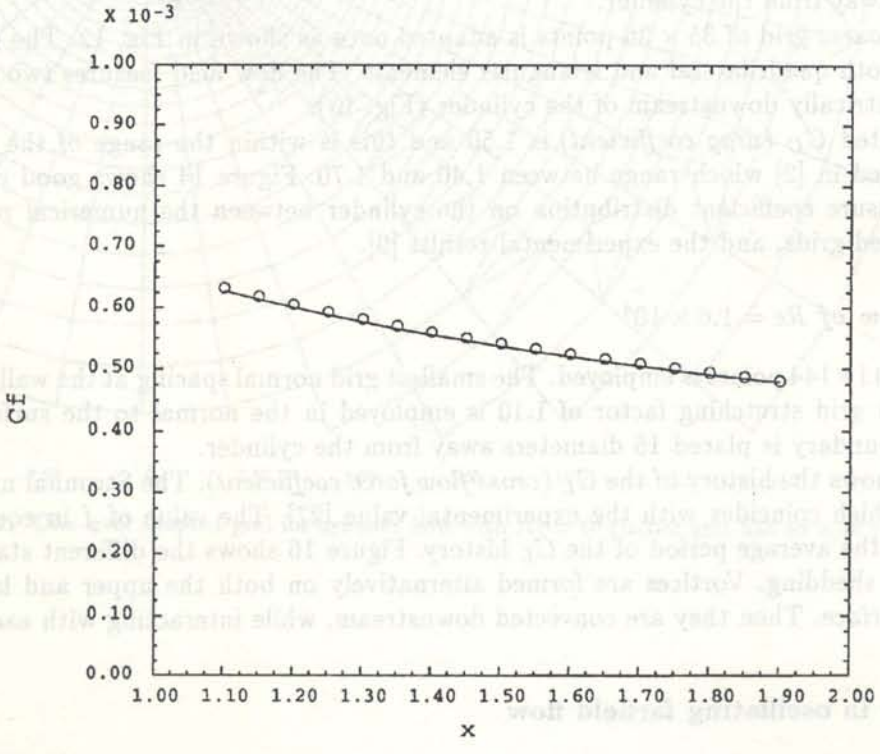


Fig. 10. Comparison of skin-friction distributions for Blasius boundary layer with $Re = 10^6$; — numerical result, o Blasius

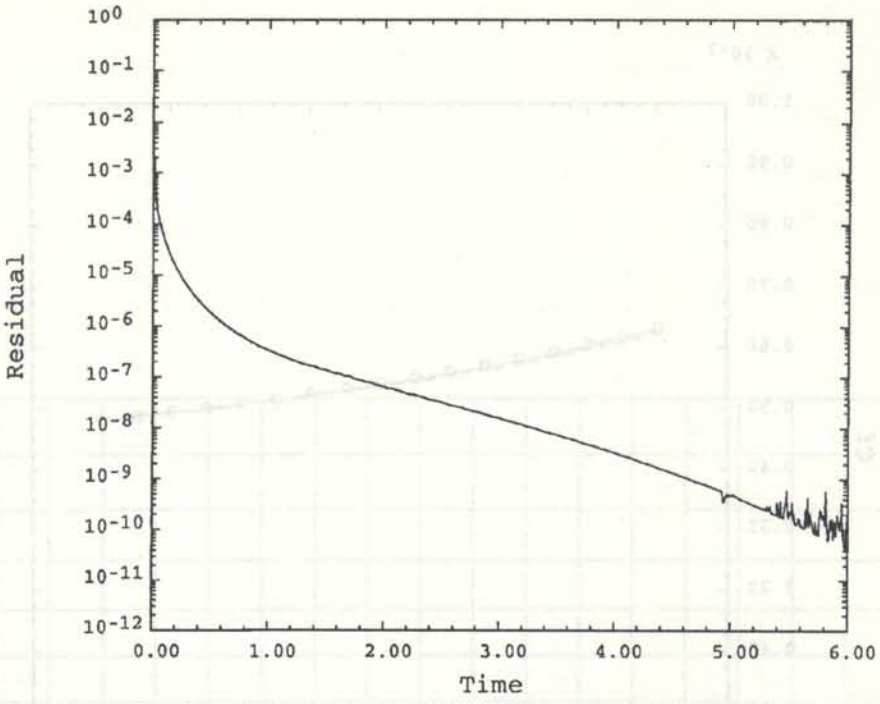


Fig. 11. Maximum residual history for Blasius boundary layer with $Re = 10^6$

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 farfield 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 flow field 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 [2] 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 [9].

Unsteady flow of $Re = 1.6 \times 10^4$

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 farfield boundary is placed 15 diameters away from the cylinder.

Figure 15 shows the history of the C_L (*cross flow force coefficient*). The Strouhal number is 0.21 in this case, which coincides with the experimental value [27]. The value of f in equation (29) 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.

5.3. Cylinder in oscillating farfield flow

This application considers flow around a circular cylinder with a planar oscillatory ambient (farfield) flow. This problem is of interest in connection with hydrodynamic forces on large cylindrical elements in offshore structures subjected to the action of waves and currents.

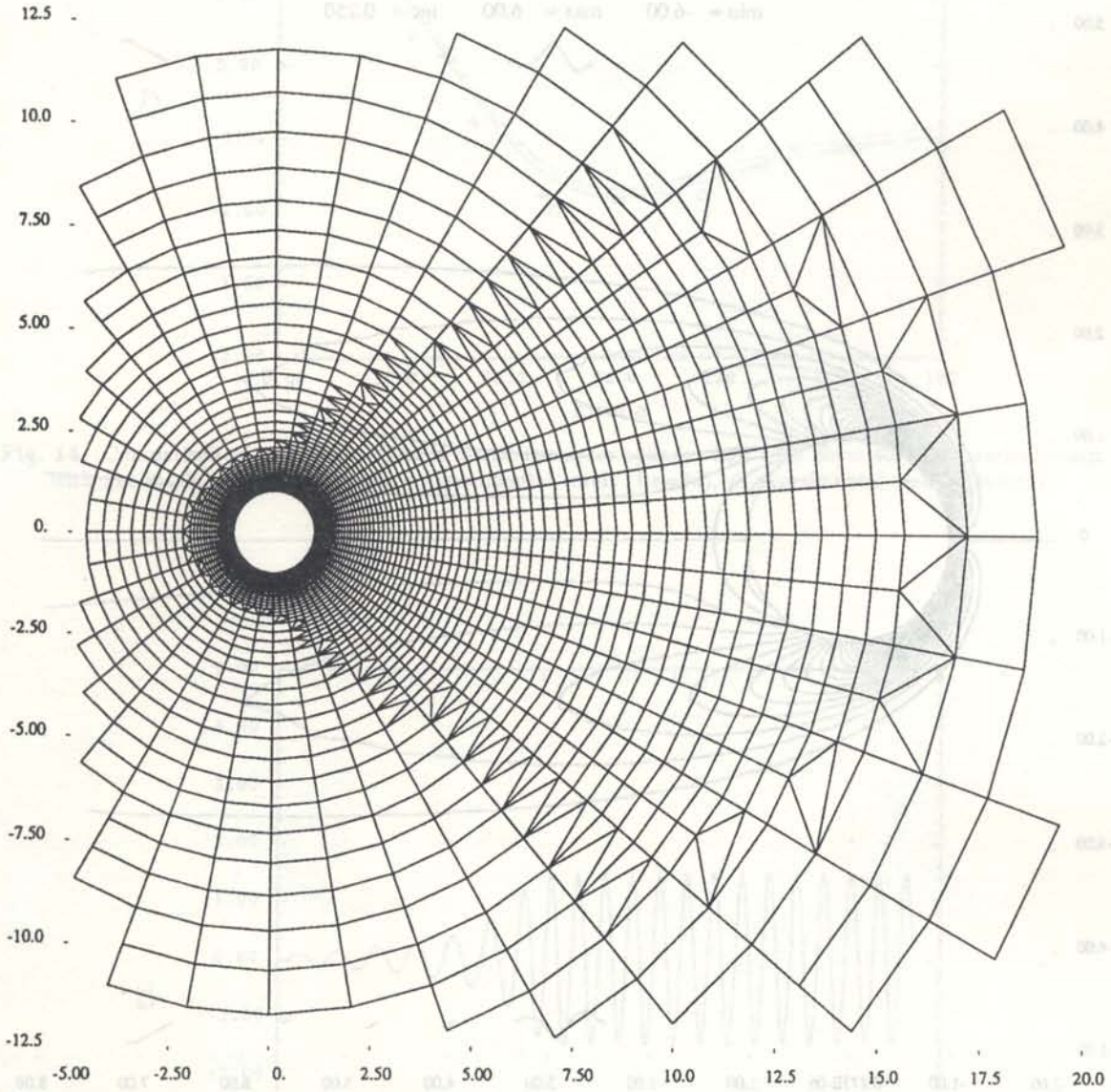


Fig. 12. One-level adapted grid for cylinder flow with $Re = 40$ (initial grid has 35×36 points)

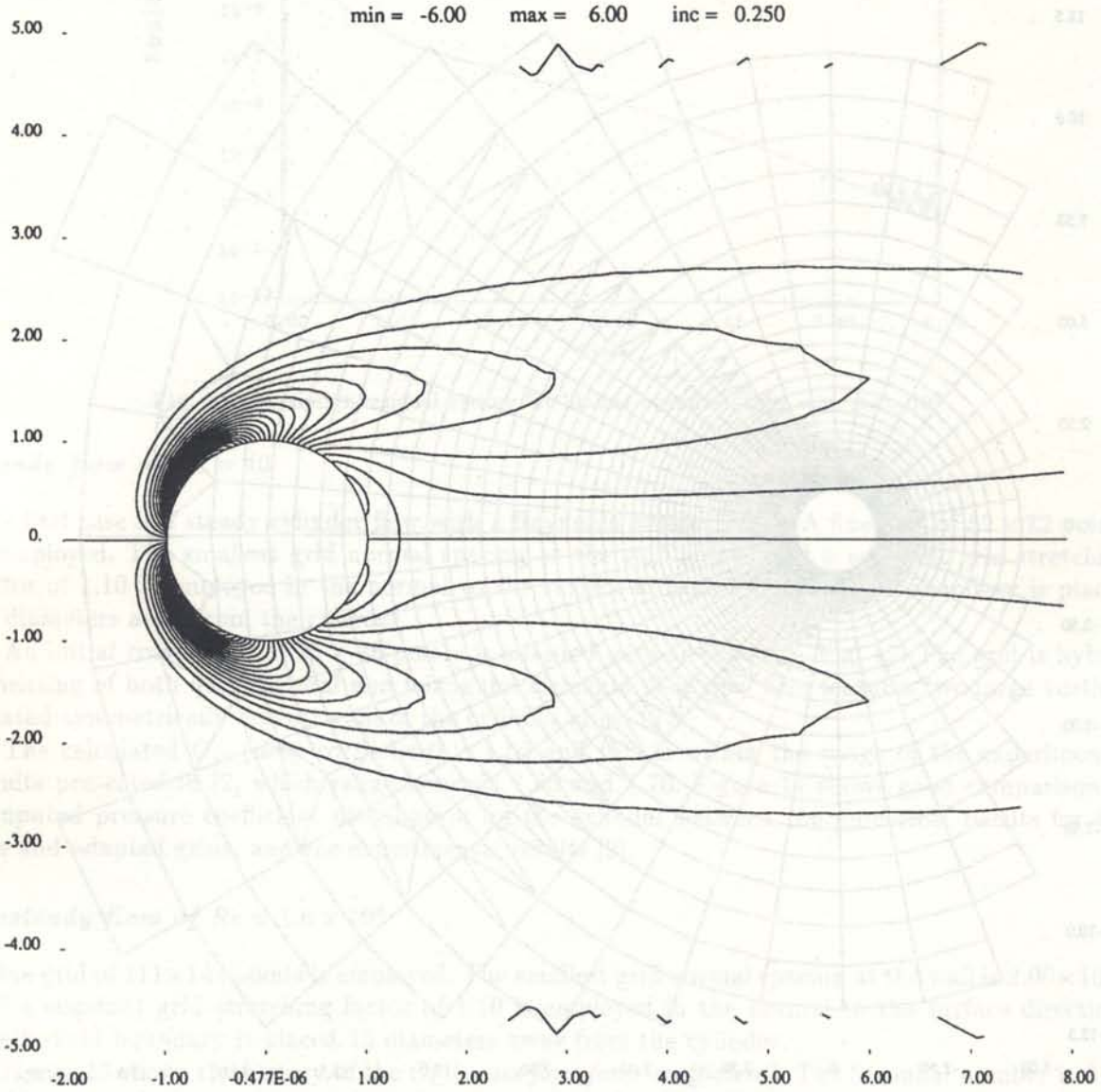


Fig. 13. Steady flow field (vorticity contours) for cylinder flow with $Re = 40$

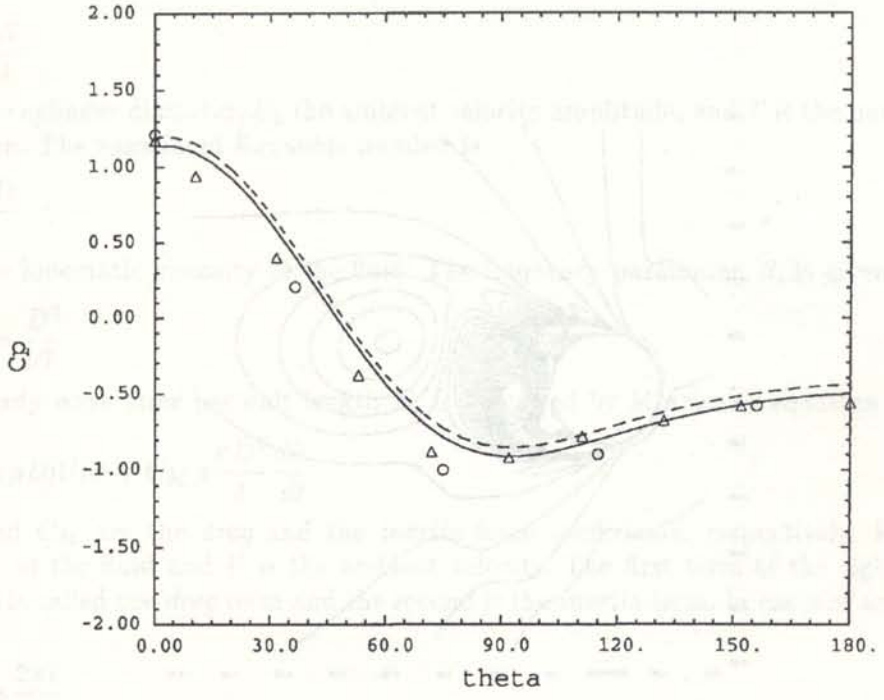


Fig. 14. Comparison of pressure coefficient distributions for cylinder flow with $Re = 40$; — numerical result with fine mesh, – – adapted mesh, o experimental result (Re=36), △ experimental result (Re=45)

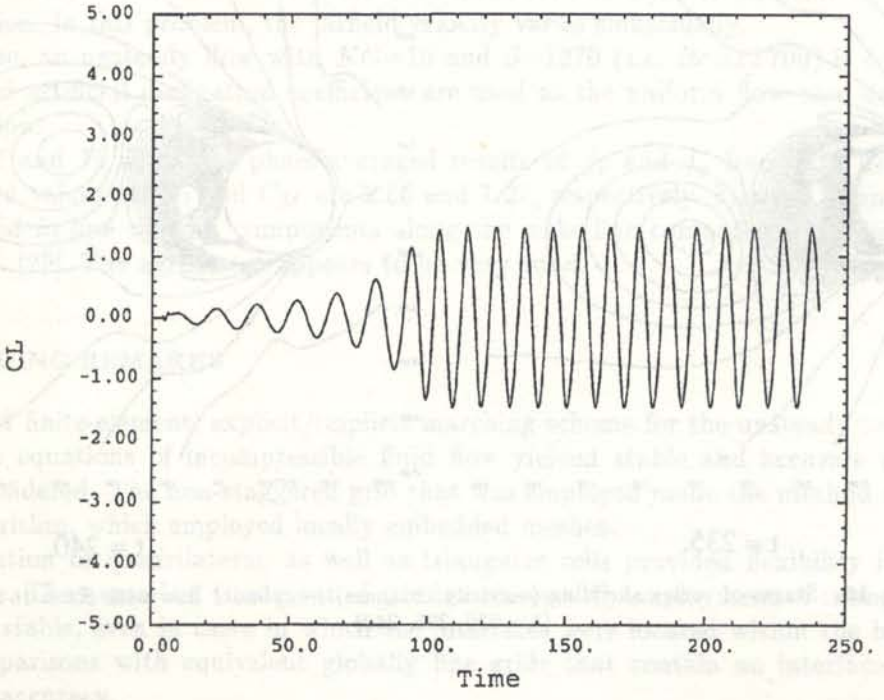
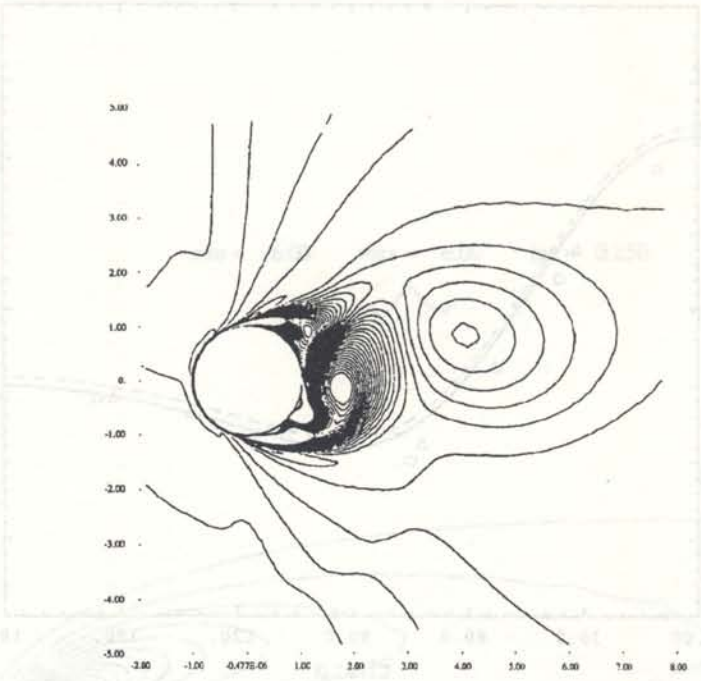
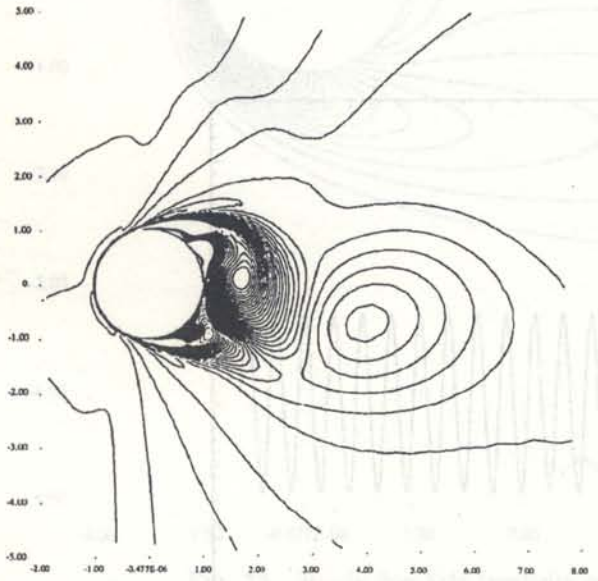


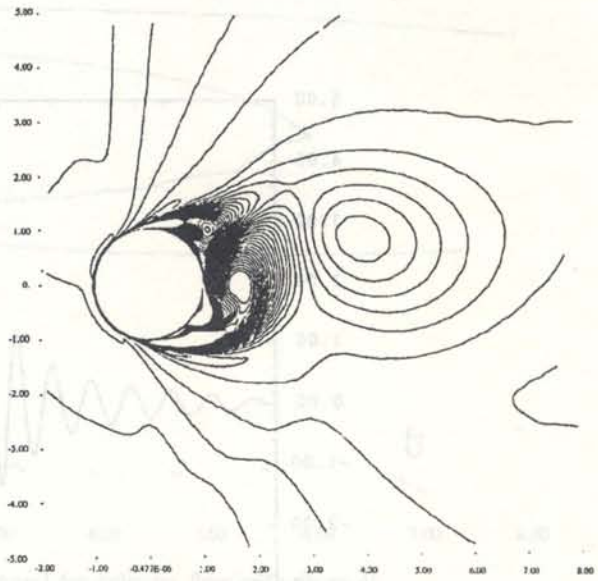
Fig. 15. Cross flow force coefficient (C_L) history — cylinder flow with $Re = 16\,200$



t = 230.



t = 235.



t = 240.

Fig. 16. Stages of vortex shedding (vorticity contours) — cylinder flow with $Re = 16\,200$ ($t = 230, 235, 240$)

For such a problem, the *Keulegan-Carpenter* number (KC) is an important parameter. It is defined by:

$$KC = \frac{U_0 T}{D}$$

where D is the cylinder diameter, U_0 the ambient velocity amplitude, and T is the period of ambient flow oscillation. The associated Reynolds number is

$$Re = \frac{U_0 D}{\nu}$$

where ν is the kinematic viscosity of the fluid. The frequency parameter, β, is given by

$$\beta = \frac{Re}{KC} = \frac{D^2}{\nu T}$$

The unsteady wave force per unit length F_x is described by Morrison's equation [27]:

$$F_x = \frac{1}{2} C_D \rho D |U| U + C_M \rho \frac{\pi D^2}{4} \frac{dU}{dt} \tag{30}$$

where C_D and C_M are the *drag* and the *inertia* force coefficients, respectively. Furthermore, ρ is the density of the fluid and U is the ambient velocity. The first term of the right hand side of equation (30) is called the *drag* term and the second is the *inertia* term. In cases of ambient velocity of the form:

$$U = U_0 \cos \frac{2\pi t}{T}$$

equation (30) can be written as follows:

$$F_x = \frac{1}{2} C_D \rho D U_0^2 \left| \cos \frac{2\pi t}{T} \right| \cos \frac{2\pi t}{T} - \frac{1}{2} C_M \frac{\rho \pi^2 D^2}{T} U_0 \sin \frac{2\pi t}{T} \tag{31}$$

In this calculation, the numerical results are compared with experimental results described in [29]. The boundary conditions are the same as those for the uniform ambient flow cases of the previous section. In this problem, the farfield velocity varies sinusoidally.

In this case, an unsteady flow with KC=10 and β=1270 (*i.e.* Re=12 700) is considered. The same grid and artificial dissipation coefficient are used as the uniform flow case described in the previous section.

Figures 17 and 18 show the phase-averaged results of f_x and f_y from $t = 6T$ to $t = 10T$. The calculated values of C_D and C_M are 2.65 and 1.27, respectively. Figures 19 and 20 show the phase-averaged *in-line* velocity components along the wake line compared with the experimental results of Ref. [29]. The agreement appears to be very good.

6. 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 non-staggered 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.

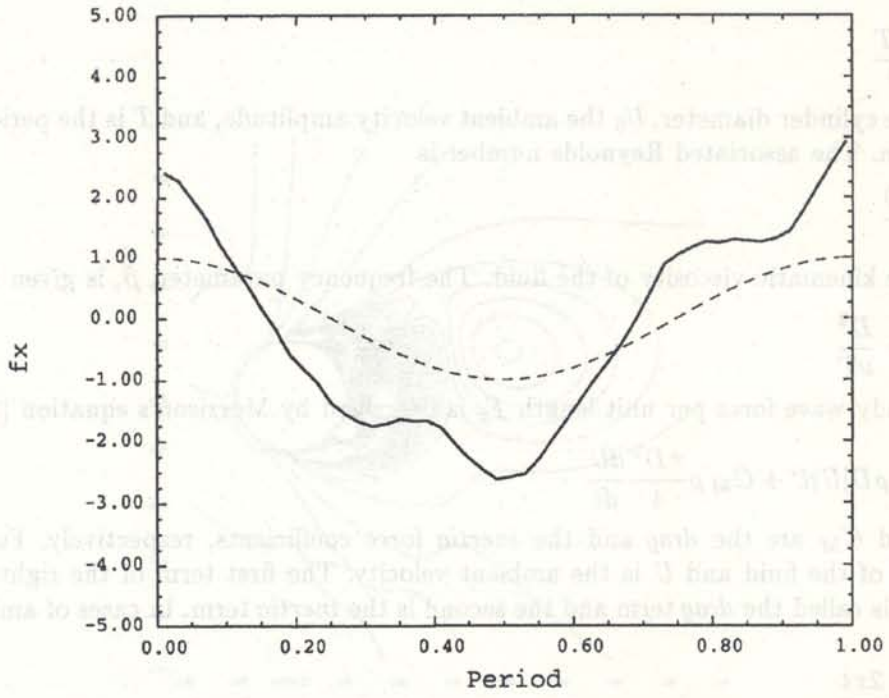


Fig. 17. Phase-averaged history of *in-line force* coefficient — oscillating flow KC=10, β=1270, Re=12 700; — *in-line force* coefficient, - - ambient velocity

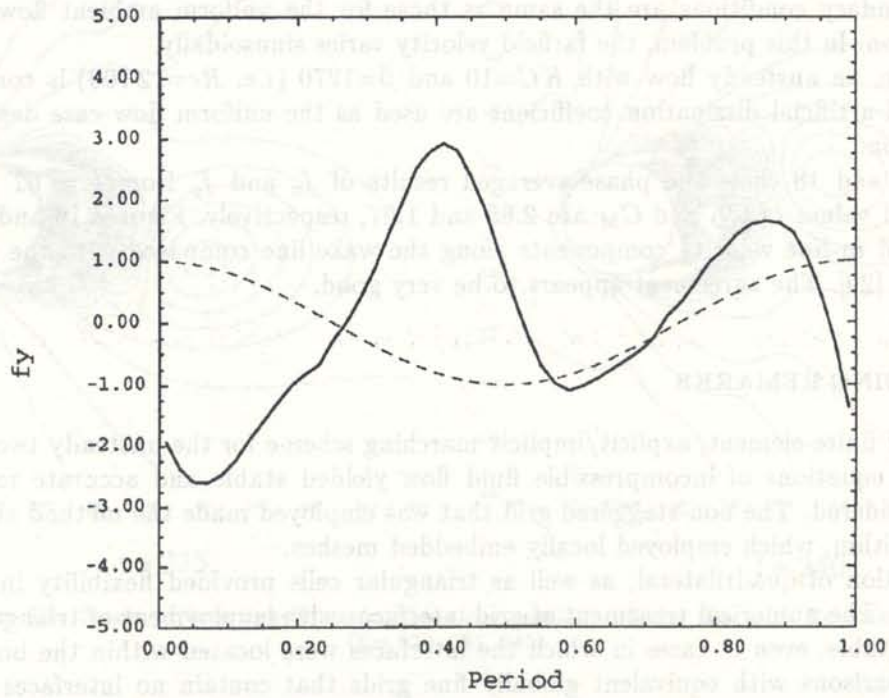


Fig. 18. Phase-averaged history of *transverse force* coefficient — oscillating flow KC=10, β=1270, Re=12 700; — *transverse force* coefficient, - - ambient velocity

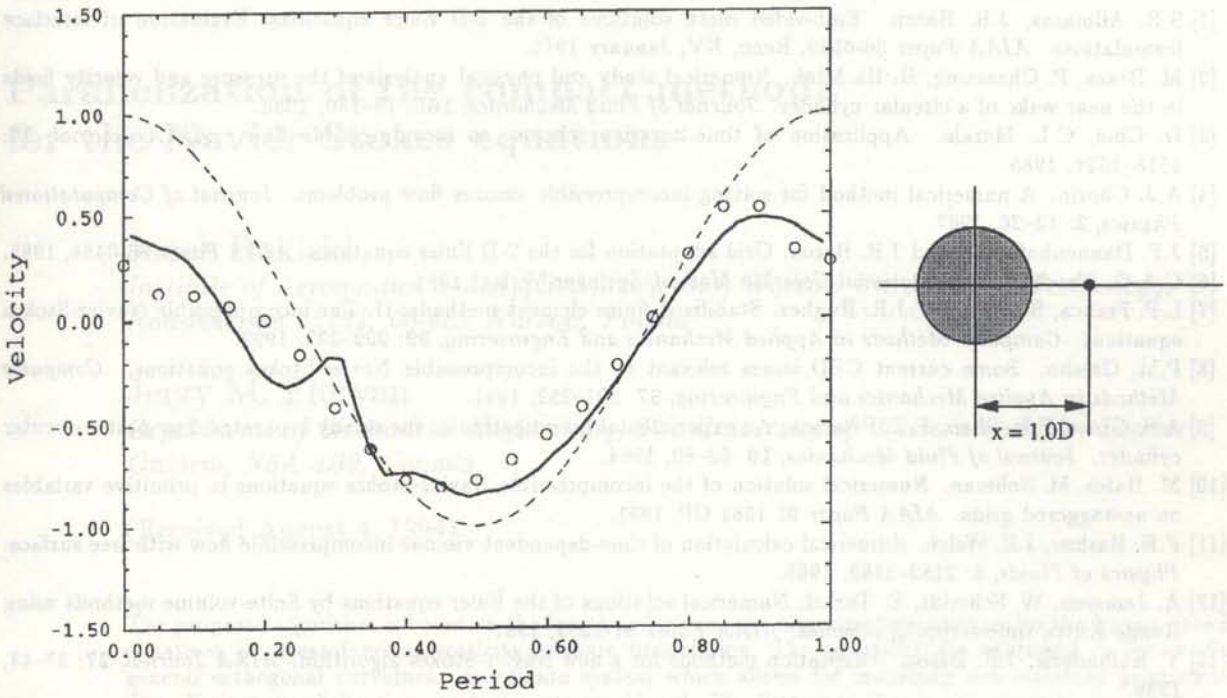


Fig. 19. Phase-averaged history of *in-line* component of velocity (u) — oscillating flow KC=10, β=1270, Re=12 700. Point (x, y)=$(1.0D, 0)$; — calculated value of u, – – ambient velocity, o experiment

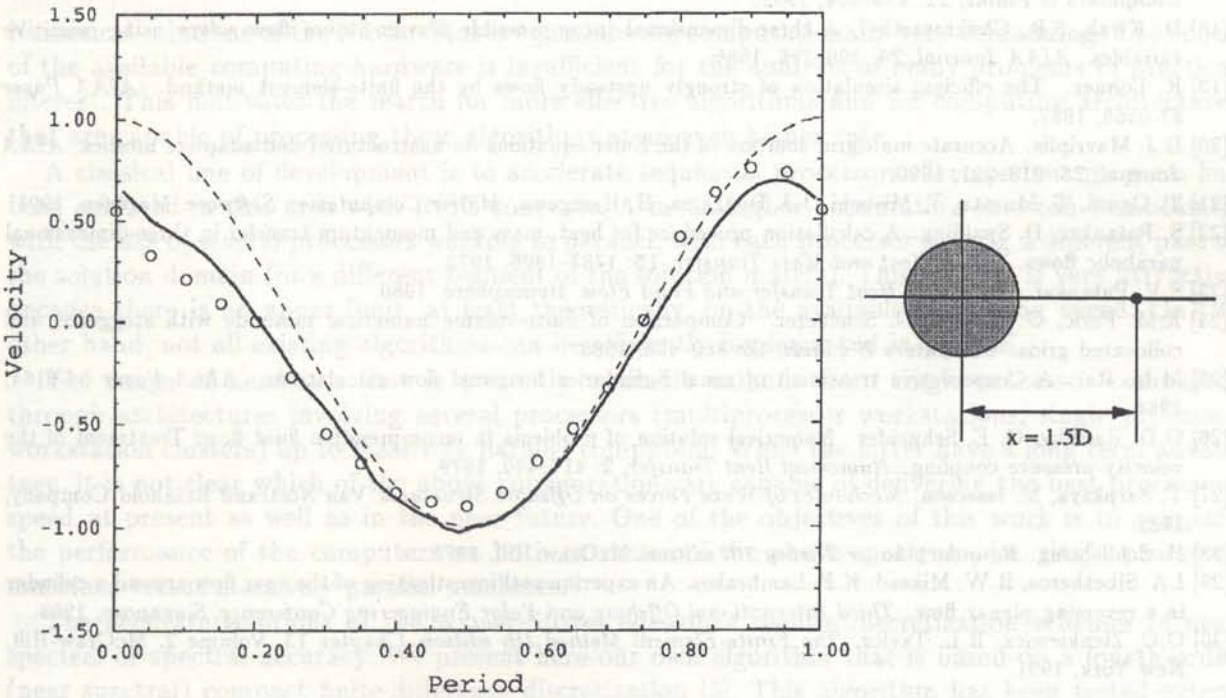


Fig. 20. Phase-averaged history of *in-line* component of velocity (u) — oscillating flow KC=10, β=1270, Re=12 700, point (x, y)=$(1.5D, 0)$; — calculated value of u, – – ambient velocity, o experiment

REFERENCES

[1] S.R. Allmaras, J.R. Baron. Embedded mesh solutions of the 2-D Euler equations: Evaluation of interface formulations. *AIAA Paper* 86-0509, Reno, NV, January 1986.

[2] M. Braza, P. Chassaing, H. Ha Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, **165**: 79–130, 1986.

[3] D. Choi, C.L. Merkle. Application of time-iterative schemes to incompressible flow. *AIAA Journal*, **23**: 1518–1524, 1985.

[4] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, **2**: 12–26, 1967.

[5] J.F. Dannenhoffer III and J.R. Baron. Grid adaptation for the 2-D Euler equations. *AIAA Paper* 85-0484, 1985.

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

[7] L.P. Franca, S.L. Frey, T.J.R. Hughes. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **99**: 209–233, 1992.

[8] P.M. Gresho. Some current CFD issues relevant to the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **87**: 201–252, 1991.

[9] A.S. Grove, F.H. Shair, E.E. Petersen. An experimental investigation of the steady separated flow past a circular cylinder. *Journal of Fluid Mechanics*, **19**: 60–80, 1964.

[10] M. Hafez, M. Soliman. Numerical solution of the incompressible Navier-Stokes equations in primitive variables on unstaggered grids. *AIAA Paper* 91-1561 CP, 1991.

[11] F.H. Harlow, J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow with free surface. *Physics of Fluids*, **8**: 2182–2189, 1965.

[12] A. Jameson, W. Schmidt, E. Turkel. Numerical solutions of the Euler equations by finite-volume methods using Runge-Kutta time-stepping schemes. *AIAA Paper* 81-1259, 1981.

[13] Y. Kallinderis, J.R. Baron. Adaptation methods for a new Navier-Stokes algorithm. *AIAA Journal*, **27**: 37–43, 1989.

[14] Y. Kallinderis. Algebraic turbulence modeling for adaptive unstructured grids. *AIAA Journal*, **30**: 631–639, 1992.

[15] Y. Kallinderis. Numerical treatment of grid interfaces for viscous flows. *Journal of Computational Physics*, **98**: 129–144, 1992.

[16] Y. Kallinderis. A finite volume Navier-Stokes algorithm for adaptive grids. *International Journal for Numerical Methods in Fluids*, **15**: 193–217, 1992.

[17] S.W. Kim, T.J. Benson. Comparison of the SMAC, PISO and iterative-advancing schemes for unsteady flows. *Computers & Fluids*, **21**: 435–454, 1992.

[18] D. Kwak, S.R. Chakravarthy. A three-dimensional incompressible Navier-Stokes flow solver using primitive variables. *AIAA Journal*, **24**: 390–396, 1986.

[19] R. Löhner. The efficient simulation of strongly unsteady flows by the finite-element method. *AIAA Paper* 87-0555, 1987.

[20] D.J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal*, **28**: 213–221, 1990.

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

[22] S. Patankar, D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat and Mass Transfer*, **15**: 1787–1806, 1972.

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

[24] R.M. Perić, G. Kessler, G. Scheuerer. Comparison of finite-volume numerical methods with staggered and collocated grids. *Computers & Fluids*, **16**: 389–403, 1988.

[25] M.M. Rai. A Conservative treatment of zonal boundaries for zonal flow calculations. *AIAA Paper* 84-0164, 1984.

[26] G.D. Raithby, G. E. Schneider. Numerical solution of problems in incompressible fluid flow: Treatment of the velocity-pressure coupling. *Numerical Heat Transfer*, **2**: 417–440, 1979.

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

[28] H. Schlichting. *Boundary-Layer Theory 7th edition.* McGraw-Hill, 1979.

[29] I.A. Sibetheros, R.W. Miksad, K.F. Lambrakos. An experimental investigation of the near flow around a cylinder in a reversing planar flow. *Third International Offshore and Polar Engineering Conference*, Singapore, 1993.

[30] O.C. Zienkiewicz, R.L. Taylor. *The Finite-Element Method 4th edition*, Chapter 13, Volume 2. McGraw-Hill, New York, 1991.