INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **50**:199–218
Published online 15 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1050

Two-dimensional implicit time-dependent calculations on adaptive unstructured meshes with time evolving boundaries

Paul T. Lin[1,*,†], Timothy J. Baker[2], Luigi Martinelli[2] and Antony Jameson[3]

[1] *Sandia National Laboratories, P.O. Box 5800 MS 0316, Albuquerque, NM 87185-0316, U.S.A.*
[2] *Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, U.S.A.*
[3] *Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, U.S.A.*

SUMMARY

An implicit multigrid-driven algorithm for two-dimensional incompressible laminar viscous flows has been coupled with a solution adaptation method and a mesh movement method for boundary movement. Time-dependent calculations are performed implicitly by regarding each time step as a steady-state problem in pseudo-time. The method of artificial compressibility is used to solve the flow equations. The solution mesh adaptation method performs local mesh refinement using an incremental Delaunay algorithm and mesh coarsening by means of edge collapse. Mesh movement is achieved by modeling the computational domain as an elastic solid and solving the equilibrium equations for the stress field. The solution adaptation method has been validated by comparison with experimental results and other computational results for low Reynolds number flow over a shedding circular cylinder. Preliminary validation of the mesh movement method has been demonstrated by a comparison with experimental results of an oscillating airfoil and with computational results for an oscillating cylinder. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: solution adaptation; moving boundaries; unstructured mesh; artificial compressibility; time-dependent calculation

*Correspondence to: Paul T. Lin, Sandia National Laboratories, P.O. Box 5800 MS 0316, Albuquerque, NM 87185-0316, U.S.A.
†E-mail: ptlin@sandia.gov

Contract/grant sponsor: Office of Naval Research (ONR) Augmentation Award for Science and Engineering Research Training (AASERT)
Contract/grant sponsor: United States Department of Energy's National Nuclear Security Administration; contract/grant number: DE-AC04-94AL85000
Contract/grant sponsor: Princeton University Department of Mechanical and Aerospace Engineering Guggenheim Foundation

Received 25 January 2005
Revised 29 April 2005
Accepted 10 May 2005

INTRODUCTION

Time-dependent phenomena are important in a variety of compressible and incompressible flows. Examples include free surface flows, turbulent flows, unsteady wake flows, and time-dependent geometries such as helicopter rotor blades, compressor and turbine blades, as well as flows with objects in relative motion such as the store separation problem. Accurate and efficient numerical simulation of these types of problems will typically involve the combination of a fast and efficient flow solver, solution adaptation, and a mesh movement algorithm for the moving boundaries.

For the calculation of time-dependent flows, implicit methods offer the advantage over explicit methods of allowing the size of the time step be dictated by the physics of the flow rather than by stability considerations. Jameson [1] proposed a fast and efficient way to calculate time-dependent flows using a multigrid-driven dual time stepping scheme. This method was improved by Melson *et al.* [2] and has been used successfully in the study of aeroelastic problems [3], low Reynolds number wake flows [4], for the calculation of free surface flows with plunging breakers [5], and for the calculation of high speed compressible flows [6]. This method has also been applied to both structured meshes [1–4] and unstructured meshes [5–9].

In a true incompressible flow the acoustic speed is infinite. The disparity in the acoustic and convective wave speeds makes the system of equations ill-conditioned. A preconditioning matrix can be introduced to reduce this disparity in the wave speeds. Chorin [10] proposed the method of artificial compressibility for steady state problems, which was later modified to treat time-dependent problems [11–13]. This artificial compressibility approach has also been used successfully with Jameson's implicit multigrid-driven dual time stepping method [4, 5, 8].

For many flow problems, there tends to be a large disparity in length scales which leads to meshes with a comparable disparity in mesh size. One of the problems of generating a grid that requires a large variation in mesh size is that one needs some *a priori* knowledge of the solution in order to place grid points where there will be high gradients. However, one usually does not know *a priori* where regions of fine resolution are needed, and for the case of a time-dependent problem, there is the additional problem that regions where fine resolution is required can change position [14]. Lack of *a priori* knowledge can lead to inefficient placement of mesh points or, even worse, insufficient resolution of the flow field if certain important flow features are not captured because the mesh is too coarse in that region. A solution adaptation approach tends to be more general as well as more practical than trying to construct *ad hoc* meshes which require educated guesses as to where the enriched or coarse parts of the mesh should be. For shocks propagating across a domain, for example, the alternative to mesh adaptation would be a uniformly fine mesh.

Solution adaptation can be achieved by three different types of modification: changing the order of accuracy of the discretized approximation (p-refinement), changing the number of points by insertion and deletion (h-refinement), and by rearranging the distribution of points (r-refinement). The last two types of modification involve altering the mesh. Often more than one type of refinement will be used. The two main approaches of h-refinement for unstructured meshes of triangles in two dimensions or tetrahedra in three dimensions are by the insertion of new points on the edges [14–20] and by insertion of new points by a Delaunay based method [6, 21–24]. The former approach offers the advantage of being straightforward to implement.

The latter approach is more flexible because it allows points to be inserted anywhere rather than being constrained to insertion on an edge [25].

Many different approaches for handling boundary movement have been proposed by numerous authors [5, 6, 9, 24, 26–34]. The simplest approach is to regenerate the mesh, but this can be quite expensive. Other methods open up pockets and remesh, but they are often limited in their degree of robustness. Various methods have been proposed that maintain the connectivity of the grid during mesh movement. The displacement vector between the points in the grid at time t and time $t+\Delta t$ is defined as $\mathbf{u}=(u_1,u_2)$ where $u_1=x_1'-x_1$, $u_2=x_2'-x_2$ and (x_1,x_2) are the coordinates of the point in the mesh at time t while (x_1',x_2') are the corresponding coordinates for its new location at time $t+\Delta t$. The displacement on the boundary is known but it is necessary to solve for the displacement in the interior. One approach that maintains the connectivity of the grid during mesh movement is to solve the Laplace equations $\nabla^2 u_1=0$ and $\nabla^2 u_2=0$ subject to the boundary conditions that u_1 and u_2 are specified on the boundary. Another approach is to use a tension spring analogy to allow the grid points to react to boundary movement [9, 26, 27]. Here, the spring stiffness for a given edge is inversely proportional to a certain power of the edge length. A third approach is to model the domain as an elastic solid and solve the equilibrium equations for the stress field [31, 32, 35, 36].

After extensive mesh movement (i.e. r-refinement) the mesh will become so deformed that any further mesh movement will cause some edges to cross-over leading to the formation of inverted cells. At this point, one can regenerate the mesh in its entirety or remesh a pocket of space that surrounds the badly distorted region. A valid retriangulation can always be generated in the planar case but there is no such guarantee in 3D space, leading to the possibility that a 3D implementation of a remeshing procedure could fail to work. Alternative treatments based on overset meshes are popular for problems involving boundary displacement since these methods can handle moving geometries with relative ease. Their accuracy, however, is suspect since the solution variables must be interpolated across the overlap region to couple the solutions in the different overset meshes.

The approach adopted in this paper employs a mesh modification algorithm based on coarsening the r-refined mesh to remove the cells that have become badly distorted and then enriching the coarsened mesh to recover a new mesh that conforms with the displaced geometry and has a quality comparable to that of the original mesh. This dynamic adaptation strategy [37] allows the distorted mesh to transition through a series of states until the final, good quality mesh has been established. A particular advantage of our approach lies in the fact that at every stage throughout the transition there is a valid, conforming and non-overlapping mesh through the entire meshed domain. This leads to a highly robust algorithm for handling moving boundary problems in both two and three dimensions.

The work described in this paper couples the fast and efficient multigrid-driven implicit method of Jameson [1] and the method of artificial compressibility of Chorin [10] with an extension of the h-refinement solution mesh adaptation and mesh movement approach for moving boundaries of Baker [36]. This mesh adaptation approach uses a Delaunay based coarsening and refinement technique for flexibility and to maintain good grid quality. The Voronoï segment method of Rebay [38] is used for enrichment. The mesh movement algorithm maintains the connectivity of the grid during mesh movement and models the domain as an elastic solid by solving the equilibrium equations for the stress field to obtain the displacement vector of the grid points. A decision on whether deletion or enrichment should occur is based on how far the error indicator is above or below the mean. This procedure offers a first step towards a more automated way of deciding which points should be deleted or inserted rather than requiring the user to supply the upper and lower thresholds for an error indicator or detector.

One advantage of using a dual time stepping approach with mesh adaptation is that because mesh adaptation occurs during each pseudo-transient calculation to determine the solution for the next physical time step, there is no time lag between the adapted mesh and the flow solution in physical time.

To validate the flow solver and the solution adaptation approach, and to demonstrate how this method can be used to study wake flows, numerical computations have been performed for unsteady incompressible flow around a cylinder for the Reynolds number range of $50 \leqslant Re \leqslant 175$ and compared with both experimental results and other computational results. Preliminary validation of the mesh movement method has been performed for two test cases. The dynamic lift over an oscillating NACA 0012 airfoil in inviscid flow is compared for the case where a nondeforming grid is rigidly attached to the airfoil and the case where the grid is allowed to deform as the airfoil oscillates. The second test case is a cylinder in a low Reynolds number flow ($Re = 100$) that undergoes a forced oscillation along the direction of the flow. This case demonstrates the ability of the mesh movement and adaptation algorithm to handle a moving body in conjunction with the flow adaptation procedure that adapts to the location of the vortices. Although the work described in this paper focuses on the computation of incompressible flows, these techniques are applicable to compressible flows.

NUMERICAL DISCRETIZATION

The governing equations of concern are the time-dependent incompressible Navier–Stokes equations in two dimensions. We will be considering laminar, constant viscosity flow without body forces. For a homogeneous incompressible flow, density is constant and will be nondimensionalized to unity.

The governing equations in integral form for an arbitrary mesh with arbitrary mesh velocity $(u_{\text{mesh}}, v_{\text{mesh}})$ are

$$\frac{\mathrm{d}}{\mathrm{d}t} \iint_{\Omega} \mathbf{z}\,\mathrm{d}x\,\mathrm{d}y + \oint_{\partial\Omega} [(\mathbf{f} - \mathbf{f}_{\mathrm{v}})\,\mathrm{d}y - (\mathbf{g} - \mathbf{g}_{\mathrm{v}})\,\mathrm{d}x] = 0 \tag{1}$$

where $\mathbf{z}$ is the vector

$$\mathbf{z} = \begin{pmatrix} 1 \\ u_1 \\ u_2 \end{pmatrix}$$

and $\mathbf{f}$ and $\mathbf{g}$ are the Euler flux vectors and $\mathbf{f}_{\mathrm{v}}$ and $\mathbf{g}_{\mathrm{v}}$ are the viscous flux vectors

$$\mathbf{f} = \begin{pmatrix} (u - u_{\text{mesh}}) \\ u(u - u_{\text{mesh}}) + p \\ v(u - u_{\text{mesh}}) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} (v - v_{\text{mesh}}) \\ u(v - v_{\text{mesh}}) \\ v(v - v_{\text{mesh}}) + p \end{pmatrix}, \quad \mathbf{f}_{\mathrm{v}} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \end{pmatrix}, \quad \mathbf{g}_{\mathrm{v}} = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \end{pmatrix}$$

Ω represents the control volume and $\partial\Omega$ represents the boundary of the control volume. The dimensionless viscous stress tensor components are

$$\tau_{xx} = \frac{1}{Re}\left(2\frac{\partial u}{\partial x}\right), \quad \tau_{xy} = \tau_{yx} = \frac{1}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \quad \tau_{yy} = \frac{1}{Re}\left(2\frac{\partial v}{\partial y}\right)$$

where $Re = U_\infty L/v$ is the Reynolds number, v is the kinematic viscosity, and U_∞ and L are the characteristic velocity and the length scales of the problem.

The governing equations are discretized by a finite volume scheme on unstructured triangular meshes. The flow variables are stored at the nodes and the control volumes are the nonoverlapping polygons that surround the nodes. The contour integrals of the convective and viscous terms are computed using the trapezoidal rule. This approximation of the spatial term can be shown to be equivalent to a Galerkin finite-element discretization with linear elements under certain conditions [9, 39, 40].

The flow variables in the first integral of Equation (1), however, are assumed to be essentially constant over each control volume and Equation (1) is now replaced by the semi-discrete equation

$$\frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{z}V] + \mathbf{R}(\mathbf{w}) = 0$$

where $\mathbf{R}(\mathbf{w})$ is the sum of the convective and viscous fluxes and the artificial dissipation. The variable $\mathbf{w}$ is defined as the vector $\mathbf{w} = (p, u_1, u_2)$.

Let $\mathbf{w}^n$ be the vector whose components are the values of $\mathbf{w}$ at each mesh point at time $t = n\Delta t$ where Δt is the physical time step. The solution for $\mathbf{w}^{n+1}$ is obtained by an implicit solution of the equation

$$\mathbf{R}^*(\mathbf{w}^{n+1}) \equiv \frac{\mathrm{d}}{\mathrm{d}t}[\mathbf{z}^{n+1}V^{n+1}] + \mathbf{R}(\mathbf{w}^{n+1}) = 0$$

In this work, $\mathrm{d}/\mathrm{d}t$ is discretized using a second or third-order accurate backward difference formula.

The method of artificial compressibility is employed to both convert the hyperbolic–elliptic unsteady incompressible Navier–Stokes equations to a hyperbolic–parabolic system and to reduce the disparity in the wave speeds of the system. The advantage of converting the hyperbolic–elliptic system to a hyperbolic–parabolic system is that efficient techniques for solution of hyperbolic–parabolic systems can be used. The modified residual $\mathbf{R}^*(\mathbf{w})$ (sum of the actual residual and the discretization of the time derivative term) is multiplied by a local preconditioning matrix and a pseudo-unsteady term is added to give

$$\frac{\mathrm{d}}{\mathrm{d}t^*}[\mathbf{w}V] + \mathbf{Pr}\cdot\mathbf{R}^*(\mathbf{w}) = 0$$

where

$$\mathbf{Pr} = \begin{pmatrix} \beta^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The pseudo-unsteady term is added so that advancement to the next physical time step can be accomplished by solving a steady-state problem in pseudo-time t^*.

The coefficient β is the artificial compressibility parameter, which has been chosen to depend on the local speed and, in particular, the following form [41, 42] leads to an improvement in the rate of convergence

$$\beta^2 = \max(0.25, u_1^2 + u_2^2)$$

A modified version [2] of Jameson's [1] five-stage implicit multistage scheme is used to integrate these equations until a steady state in pseudo-time has been reached. Various convergence acceleration techniques can be used to speed up the convergence to steady state in pseudo-time. The three used in this work are multigrid, local time stepping, and residual averaging (details are provided in References [8, 43]). A more detailed account of the numerical discretization can be found in References [39, 40].

The no-slip boundary condition is enforced at solid boundaries. Non-reflecting far-field boundary conditions based on a linearized characteristics approach are imposed at the outer boundary of the domain. The formulation of these far-field boundary conditions is similar to the version used by Belov [4].

MESH GENERATION AND MODIFICATION

Mesh generation

The initial mesh generation is based on a constrained Delaunay triangulation [23, 44]. The goal is to obtain a smooth gradation in mesh generation throughout the domain such that the density of the volume mesh near the boundary surface matches the mesh size of the boundary triangulation [36]. The length density function ρ at each point is related to the target circumradius of triangles in that region. Points are first placed on the boundaries in a manner specified by the user and then an initial triangulation of the volume is generated without any interior points. The initial length density function for the boundary points is calculated as an average length of the incident boundary edges. As interior points are added, the length density function of the new point is based on an area weighting of the length density function of the three vertices of the triangle that contains the new point. Points are inserted if the actual circumradius is a certain threshold larger than the length density function. Interior points are added by the Voronoï segment method of Rebay [38]. Figure 1 gives an example of an initial mesh generated around a circular cylinder.

Solution mesh adaptation

Central to any mesh adaptation scheme are two essential requirements, a means of recognizing where extra flowfield resolution is needed for accuracy reasons or not needed for efficiency reasons and, secondly, a mechanism to alter the mesh in an appropriate manner [25].

As the goal of an adaptation scheme is to achieve error equidistribution at all the mesh points, the first requirement will be fulfilled by using a suitable choice of 'error indicator'. The error indicator used in this work is the same as the numerator of the error indicator used by Löhner [14] and Liou [6]. The denominator of their error indicators is intended to handle discontinuities, so it was not necessary for this work. Our error indicator is a second

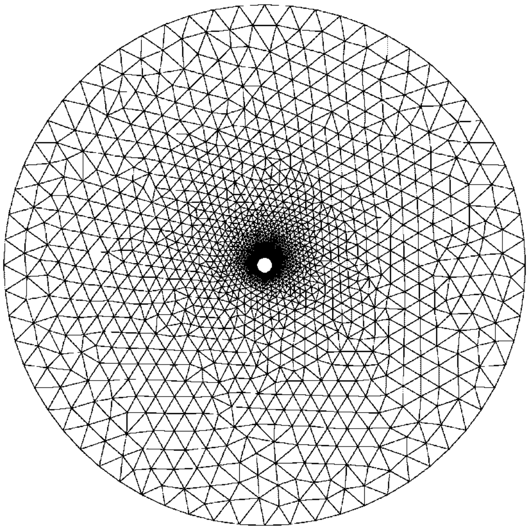

Figure 1. Unadapted grid.

derivative of an appropriate flow quantity. The second derivative is calculated by a finite volume discretization that is very similar to that used to discretize the viscous flux terms [6, 39]. Before adaptation occurs, the value of the error indicator E is calculated at each node.

Because the size of triangles is controlled by the length density function ρ, which represents a target circumradius, we need a mechanism to modify ρ. Using a modified version of the method of Baker [36], the average error indicator for the entire mesh $\bar{E}$ is computed and the the standard deviation σ is calculated from $\sigma^2 = \overline{E^2} - (\bar{E})^2$. Let

$$e_{\mathrm{up}} = \frac{E - \bar{E} - \lambda_{\mathrm{up}}\sigma}{\sigma}$$

$$e_{\mathrm{low}} = \frac{E - \bar{E} + \lambda_{\mathrm{low}}\sigma}{\sigma}$$

Then the new ρ at each mesh point becomes

$$\rho_{\mathrm{new}} = \begin{cases} \dfrac{\rho}{1 + \alpha \min(e_{\mathrm{up}}, 1)} & E > \bar{E} + \lambda_{\mathrm{up}}\sigma \\ \rho & \bar{E} - \lambda_{\mathrm{low}}\sigma \leqslant E \leqslant \bar{E} + \lambda_{\mathrm{up}}\sigma \\ \rho * [1 + \alpha \min(|e_{\mathrm{low}}|, 1)] & E < \bar{E} - \lambda_{\mathrm{low}}\sigma \end{cases}$$

λ_{up} is the fraction of a standard deviation above the mean where the length density function starts to be decreased (increasing the likelihood and amount of enrichment), λ_{low} is the fraction of a standard deviation below the mean where the length density function starts to be increased (increasing the likelihood and amount of coarsening), and α controls how rapidly changes in mesh density occur. A larger value of α will cause a larger change in mesh density. λ_{up}, λ_{low}, and α are user supplied constants. If the ratio of a triangle's circumradius to ρ_{new} is above a certain threshold, then the triangle is flagged for refinement by Delaunay point insertion with point placement determined by the Voronoï segment method of Rebay [38]. If the ratio of a triangle's circumradius to ρ_{new} falls below a certain threshold, then the triangle will be deleted.

The mechanism to alter the mesh is based on a modified version of the method of Baker [36]. Triangles that are to be deleted are removed by collapsing their shortest edge. This will actually delete two triangles. One could collapse an edge to its midpoint, to any of the endpoints, or to any location along that edge. In practice, the edge is collapsed to its midpoint, unless one of the endpoints is a boundary point. The shortest rather than the longest edge of the triangle to be removed is collapsed because one is trying to increase the average edge length locally.

After all the necessary triangles and relevant edges have been deleted, the quality of the triangulation tends to be degraded. The final mesh quality will be improved if some smoothing of the mesh is performed. This is done by swapping diagonals in order to maximize the minimum of the angles of the pair of triangles with a common edge that form a convex quadrilateral. Enrichment is accomplished by inserting a point according to the Voronoï segment method and reconstructing the mesh by an incremental Delaunay algorithm. Enrichment continues until the circumradius of each triangle reaches its ideal circumradius based on the local length density function ρ.

The example mesh in the section on initial mesh generation (Figure 1) is used as the starting point for a calculation of the incompressible flow over a circular cylinder at a Reynolds number of 150. Figure 2 shows the vorticity field for the entire domain with the mesh allowed to adapt and Figure 3 shows the adapted grid for the entire domain once periodic shedding has been established. The initial mesh had 2300 points while the adapted mesh for this example typically had around 7000–7200 points.

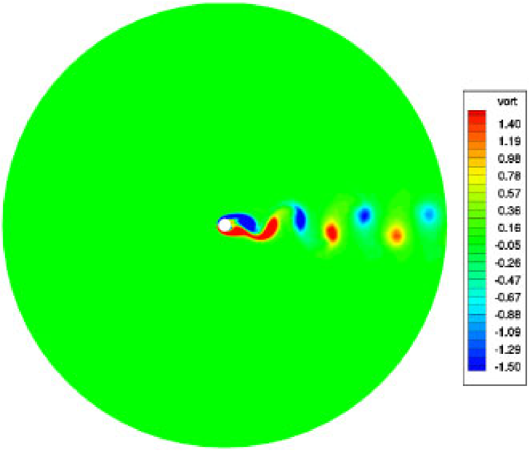

Figure 2. ω field with adaptation.

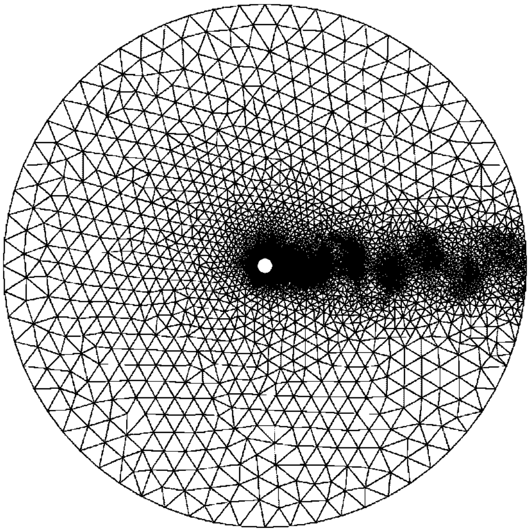

Figure 3. Adapted grid.

Mesh movement

Modification of the mesh due to the movement of boundaries for time-dependent problems follows the work of Baker [36]. This involves three stages: mesh movement, mesh coarsening, and mesh enrichment. The mesh movement is done by an r-refinement procedure. Mesh coarsening based on an edge collapse is used to remove poor quality triangles after the r-refinement stage. Mesh enrichment based on comparing the ratio of circumradius to length density function is used to form a mesh of comparable quality to the mesh before the boundary movement.

After one time step Δt the boundary of the domain is displaced to a new location. However, the interior of the mesh needs to be modified in a manner such that comparable mesh quality to the mesh before the boundary movement is attained.

The displacement vector between the points in the grid at time t and time $t+\Delta t$ is defined as $\mathbf{u}=(u_1,u_2)$ where $u_1=x_1'-x_1$ and $u_2=x_2'-x_2$. The coordinates (x_1,x_2) represent the point at time t and the new location at time $t+\Delta t$ is represented by (x_1',x_2'). For a given displacement on the boundary it is necessary to determine a suitable displacement in the interior.

Following the work of Baker [36], the equilibrium equations for the stress field will be solved. The strain tensor can be written as

$$\varepsilon_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right), \quad i,j=1,2$$

and for an isotropically elastic solid, the stress tensor can be written as

$$\sigma_{ij}=\lambda\varepsilon_{kk}\delta_{ij}+2\mu\varepsilon_{ij}, \quad i,j=1,2$$

where λ and μ are the Lamé constants. In the absence of a distributed body force, the stress field satisfies the equation

$$\frac{\partial \sigma_{ij}}{\partial x_i} = 0$$

It follows that only one user defined parameter, the ratio of the Lamé constants, needs to be specified. If desired, one can instead specify a value for Poisson's ratio,

$$v = \frac{\lambda}{2(\lambda + \mu)}$$

from which the ratio of the Lamé constants can be readily obtained.

The equation for the stress field is cast as a time marching problem to be iterated to steady state. The spatial terms are discretized by a finite volume method similar to the discretization of the viscous flux terms in the Navier–Stokes equation.

It is important to identify an appropriate set of triangles that needs to be removed during the coarsening stage. Therefore, one needs to construct suitable mesh deformation and quality measures to monitor the time evolution of the mesh. The mesh deformation and quality measures of Baker [30, 36] were used in this work.

Enrichment is accomplished by inserting a point using the Voronoï segment method and performing an incremental Delaunay algorithm. Enrichment continues until the circumradius of each triangle reaches its ideal circumradius (length density function ρ).

An example of one cycle of the mesh movement method for a pitching airfoil is shown in Figures 4–7. For clarity, we will focus on the trailing edge of the airfoil. Figure 4 shows the initial grid before the start of mesh movement. Figure 5 shows the grid after the trailing edge of the airfoil has moved down and r-refinement has taken place. The poor quality triangles produced from the r-refinement will be removed in the coarsening stage. Figure 6 shows the

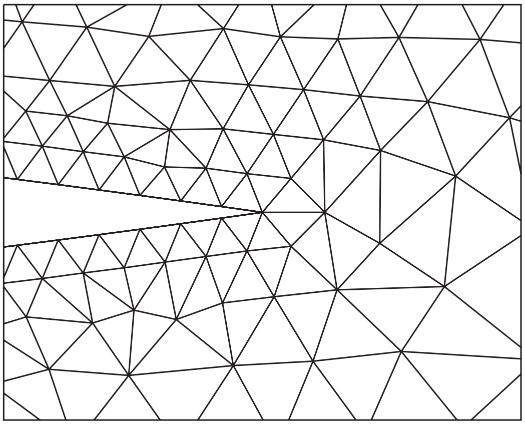

Figure 4. Initial grid near trailing edge.

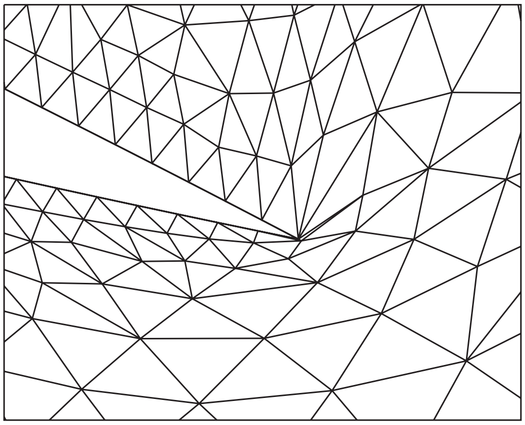

Figure 5. Mesh after rotation of airfoil.

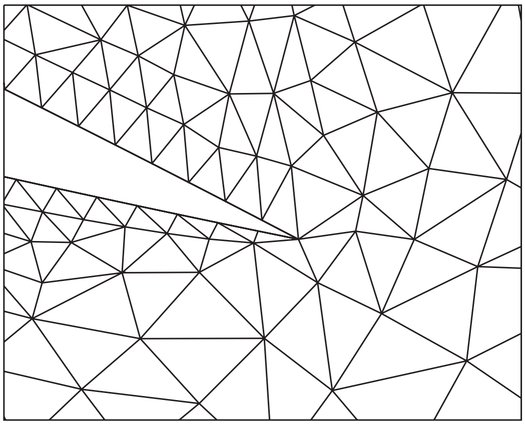

Figure 6. Coarsened mesh after rotation.

grid after the mesh coarsening and mesh smoothing operations have occurred. Figure 7 shows the grid after the mesh enrichment operation has occurred to restore the mesh point density around the trailing edge.

Generation of coarser meshes for multigrid

Mesh adaptation is performed on the finest grid only. The coarser grids for multigrid are then derived from the finest grid using a coarsener. As a first attempt for a coarsener, a fast simple coarsener that is based on initial triangulation techniques rather than edge collapsing techniques is used.

The goal of the method is analogous to the generation of coarser meshes for a vertex-based structured code in two dimensions. As in the structured mesh case, we want to roughly

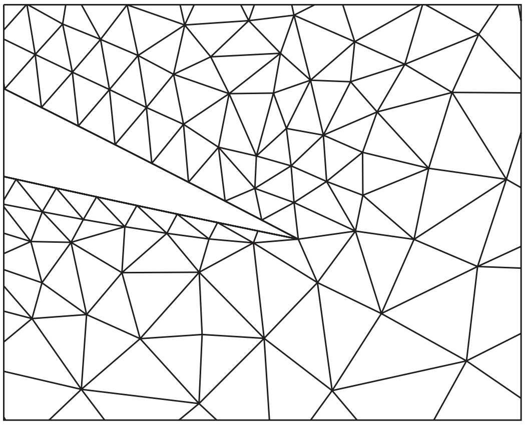

Figure 7. Complete cycle of movement and modification.

double the spacing between each point on a fine mesh in order to generate a coarser mesh. The average edge length of all edges incident to each point is calculated. This gives a rough distance measure to the neighbouring points. The coarser grid is generated first by removing every other boundary point on the fine grid and then an initial triangulation is produced. Using a Bowyer algorithm [45], an attempt is made to insert each interior point of the fine grid. A comparison is made between the distance of the potential new point to the closest existing coarse grid point and the average associated edge length of that existing coarse grid point multiplied by a factor that is a little less than 2. If the former is greater than the latter, then the potential new point becomes a coarse grid point, otherwise it is rejected.

This coarsening strategy relies on pure geometric considerations to produce a coarse mesh. An advantage is that it should produce a coarse mesh that closely resembles the fine mesh. Because the coarsener needs to be called each time after adaptation or mesh movement occurs on the finest mesh, the coarsener needs to be fast, otherwise this method will be inefficient. The key to the speed of the coarsener is a quadtree data structure [44].

Integration of mesh modification with flow solver

The details of the coupling of the flow solver with mesh movement and solution adaptation is as follows. The dual time stepping approach has an 'outer time' which advances forward one time step in physical time and an 'inner time' which is the pseudo-time for the subiterations. Iterating to steady state in pseudo-time advances forward one step in physical time. For the case where one knows the location of the boundaries in advance as a function of time, the mesh movement is called once every physical time step, before the pseudo-transient iteration. After the boundary is moved to the new location, the three stage process of r-modification, mesh coarsening, and mesh refinement occurs. Now as the pseudo-transient iteration proceeds, solution adaptation is performed, either every multigrid cycle or after a certain number of multigrid cycles [5, 6]. An example of this case is a NACA 0012 airfoil that is oscillating at a fixed or prescribed frequency.

For the case where the location of the boundary depends on the flow solution in the interior, one does not know *a priori* the location of the boundaries and the mesh movement algorithm needs to be called as the pseudo-transient iteration proceeds in conjunction with solution adaptation [5, 6]. An example of this case is a free surface flow where the location of the free surface depends on the solution of the bulk flow.

After each cycle of mesh modification, whether to adapt to the time evolving flow solution or to adapt to a relative displacement of boundary surfaces, it is necessary to project the solution onto the new mesh points. Currently, the solution vector is interpolated from the previous mesh to the current mesh with a linear interpolation. From the results shown in the next section, it is apparent that linear interpolation provided adequate accuracy for the problems we have considered. It has been noted by de Sampaio *et al.* [46] that quadratic interpolation does give much greater accuracy, which may be desirable in some situations. Our current implementation for the solution interpolation is not strictly conservative, e.g. total momentum is not conserved during mesh enrichment and mesh coarsening. This lack of conservation could be addressed if the adaptation module and solution module were closely integrated so that an interpolation that for example conserves total momentum could be done whenever a given mesh point was removed from or added to the mesh. From the results presented in this work, the lack of a strictly conservative treatment for the solution interpolation does not appear to be critical. The time integration of the flow equations is, of course, conservative and provided a modified interpolation that is conservative is used during an adaptation cycle, there will be no spurious loss or gain of any conserved quantity. This contrasts favorably with overset methods for which solution interpolation must be continually carried out as part of the time integration of the flow equations.

RESULTS AND DISCUSSION

Validation was performed with three test cases: a stationary circular cylinder in a low Reynolds number flow, an oscillating airfoil with mesh deformation in inviscid flow, and an oscillating cylinder with mesh deformation in a low Reynolds number flow. The first test case demonstrates the solution mesh adaptation approach without mesh movement. The second test case demonstrates the mesh movement approach without solution mesh adaptation. The final test case demonstrates both solution mesh adaptation and mesh movement.

One of the benchmark test cases for any unsteady incompressible laminar Navier–Stokes algorithm is flow over a circular cylinder due to the amount of experimental [47–49] and computational [4, 50] data available. Experiments conducted by Williamson [47], indicate that the flow around a circular cylinder does not become unsteady until a critical Reynolds number of about $Re_c = 49$. Spanwise independent parallel shedding is observed in the approximate range of $49 < Re < 180$, above which three-dimensional shedding modes are observed. This makes the range $49 < Re < 180$ useful for validating two-dimensional unsteady algorithms. Williamson [47] proposed a relation for Strouhal frequency, $St(Re)$, of the form $St(Re) = (A/Re) + B + C \cdot Re$ where A, B, and C are constants ($A = -3.3265$, $B = 0.1816$, $C = 1.600 \times 10^{-4}$). This relationship fits Williamson's experimental data to within about 2% [48].

Computations were performed using an initial grid that contains twice the number of body points as shown in the example grid when describing the initial mesh generation. 256 points were equally spaced around the cylinder and 64 points were equally spaced around the far field outer boundary. The total number of mesh points for the initial grid was 3200. Once periodic shedding was reached, the total number of mesh points was 9900–10 200 for the $Re = 50$ case and 14 200–15 200 for the $Re = 175$ case, with the other cases being somewhere in between. The third order backward time discretization was used, with 48 time steps per shedding cycle.

Figure 8 shows the comparison between these computational results and the experimental data. The computational results of Henderson [50] and Belov [4] are also included for comparison. Strouhal frequency is plotted on the vertical axis with Reynolds number, *Re*, on the horizontal axis. The solid line is the best fit of the experimental data to the relation $St(Re) = (A/Re) + B + C \cdot Re$. The computed Strouhal number is within 2% of the above best fit to the experimental data, and the computed *St* differs by 1% or less for the *Re* between 60 and 150. The computational results agree well with the experimental results.

Williamson and Roshko [49] also proposed a relationship of the same form but with different coefficients for the variation of base pressure coefficient $C_{p_b}(Re)$ with Reynolds number ($A = -14.3500$, $B = 0.6950$, $C = 16.920$). The base pressure coefficient is calculated from the time-averaged value of the pressure at the point 180° from the forward stagnation point. These coefficients were obtained from the work of Henderson [50]. Figure 9 shows the comparison between the above best fit to the experimental data and the computational results. Once again, the computational results of Henderson [50] and Belov [4] are also included for comparison. Base pressure coefficient is plotted on the vertical axis with Reynolds number, *Re*, on the horizontal axis. The solid line is the best fit of the experimental data to the relation $St(Re) = (A/Re) + B + C \cdot Re$. Our computed base pressure is within 5% of the best fit curve to the experimental data and is consistently lower.

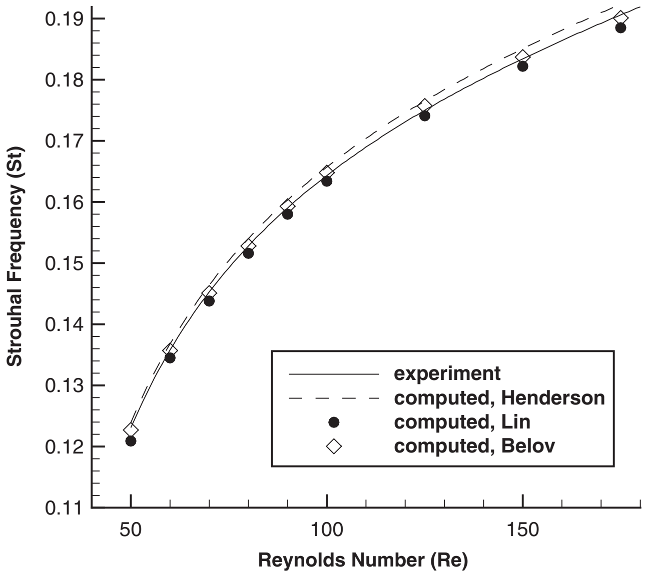

Figure 8. $St(Re)$ for cylinder $49 < Re < 180$.

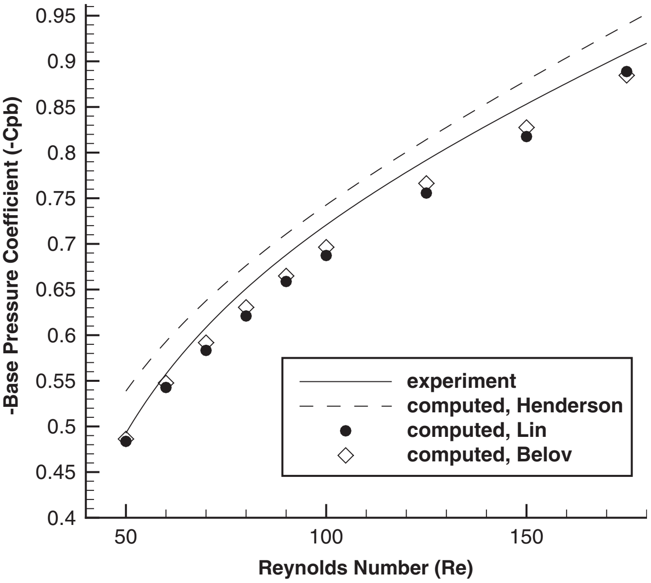

Figure 9. $-C_{\mathrm{p_b}}(Re)$ for cylinder $49 < Re < 180$.

For the second test case, the incompressible Euler equations rather than the incompressible laminar Navier–Stokes equations were solved. The test case is a NACA 0012 airfoil undergoing a forced oscillation at a reduced frequency of 0.400 for angles of attack between $-6.7°$ and $6.7°$ about a mean angle of attack of zero. The reduced frequency is defined as $\omega_r = c\omega/2U$ where ω is the frequency of oscillation, c is the chord of the airfoil, and U is the characteristic velocity (freestream velocity for our case). The airfoil is pitched about a point that is 0.37 chords downstream of the leading edge.

Two computations were performed for this test case. The first computation treats the entire grid as a rigid body that is fixed to the airfoil and pitches with the airfoil. The computational mesh has 128 points on the body of the airfoil and 64 points on the far field boundary. The total number of grid points is roughly 3000. The third-order backward discretization of the time derivative term with 48 time steps per oscillation cycle was used. The second computation has the outer boundary of the grid fixed and allows the interior of the mesh to deform as the airfoil pitches. The mesh movement procedure was used to allow the mesh to deform as the airfoil pitched. The initial mesh is the same mesh as used in the nondeforming mesh case. As in the previous case, the third-order backward discretization of the time derivative term with 48 time steps per oscillation cycle was used.

Figure 10 shows a comparison of the computed lift coefficient (C_l) vs angle of attack (α) for the two computed results with the experimental results of Halfman [51]. The experimental Reynolds number of 10^6 and Mach number of 0.1 are high and low enough, respectively, to justify comparison with a numerical solution of the incompressible Euler equations. Note that the values for lift coefficient for the deforming mesh (solid circles) and the rigid mesh (hollow squares) show good agreement, and they both agree well with the experimental results. The purpose of this test case is to demonstrate that the mesh deformation approach gives reasonable

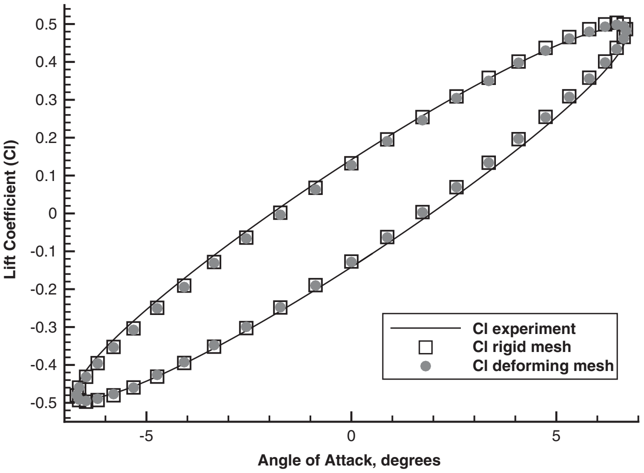

Figure 10. Comparison of C_l between rigid mesh, deforming mesh, and experiment.

results and would be appropriate for situations with multiple bodies in relative motion. An example of this situation is a multi-element airfoil with flaps and slats that are being deployed.

As a first demonstration of the ability of the method to handle problems with both time varying boundaries as well as flow solution adaptation, we consider the test case calculated by Mittal and Tezduyar [33]. They considered a circular cylinder at Reynolds number of 100 that is forced to oscillate horizontally. The forced horizontal oscillation test case is interesting because in contrast to flow past a stationary cylinder at $Re = 100$ which leads to classical unsymmetrical vortex shedding, if the frequency of the forcing is high enough, symmetric vortex shedding is observed [52]. The prescribed displacement of the cylinder (nondimensionalized by cylinder diameter) is $x(t) = 0.5(1 - \cos \omega t)$. The reduced frequency of oscillation $\omega_r = 0.35$, where reduced frequency is defined as $\omega_r = (1/2\pi)(\omega D/U)$, D and U are the diameter of the cylinder and characteristic velocity of the flow respectively. The initial condition is the periodic flow past a fixed cylinder at $Re = 100$.

We first calculated flow past a fixed cylinder at $Re = 100$ to obtain the periodic solution, then at time $t = 119.2$, started the forced oscillation of the cylinder. The initial grid had a far field 16.7 cylinder diameters from the centre of the cylinder, with 128 points on the surface of the cylinder and 64 points on the outer boundary. The total number of points in the initial grid was about 2300. The third-order backward difference discretization in time was used, with 48 time steps per period of forced oscillation. In contrast to $Re = 100$ flow past a fixed cylinder which has a lift coefficient with an amplitude of about 0.3 and a drag coefficient of about 1.3, the lift coefficient for the forced oscillation case is decaying to zero and the drag coefficient has a far larger amplitude. The calculated drag coefficient oscillates with a reduced frequency of 0.35. Our drag coefficient which is approximately between $[-2.6, 5.8]$ is similar to Mittal and Tezduyar's drag coefficient which was approximately between $[-3.1, 6.3]$.

Figure 11 shows a sequence of frames over one period of oscillation. The left column shows the vorticity field and the right column shows the corresponding grid. Each row represents $\frac{1}{4}$ of a period later. The first row is for a time when the cylinder is located at the mean

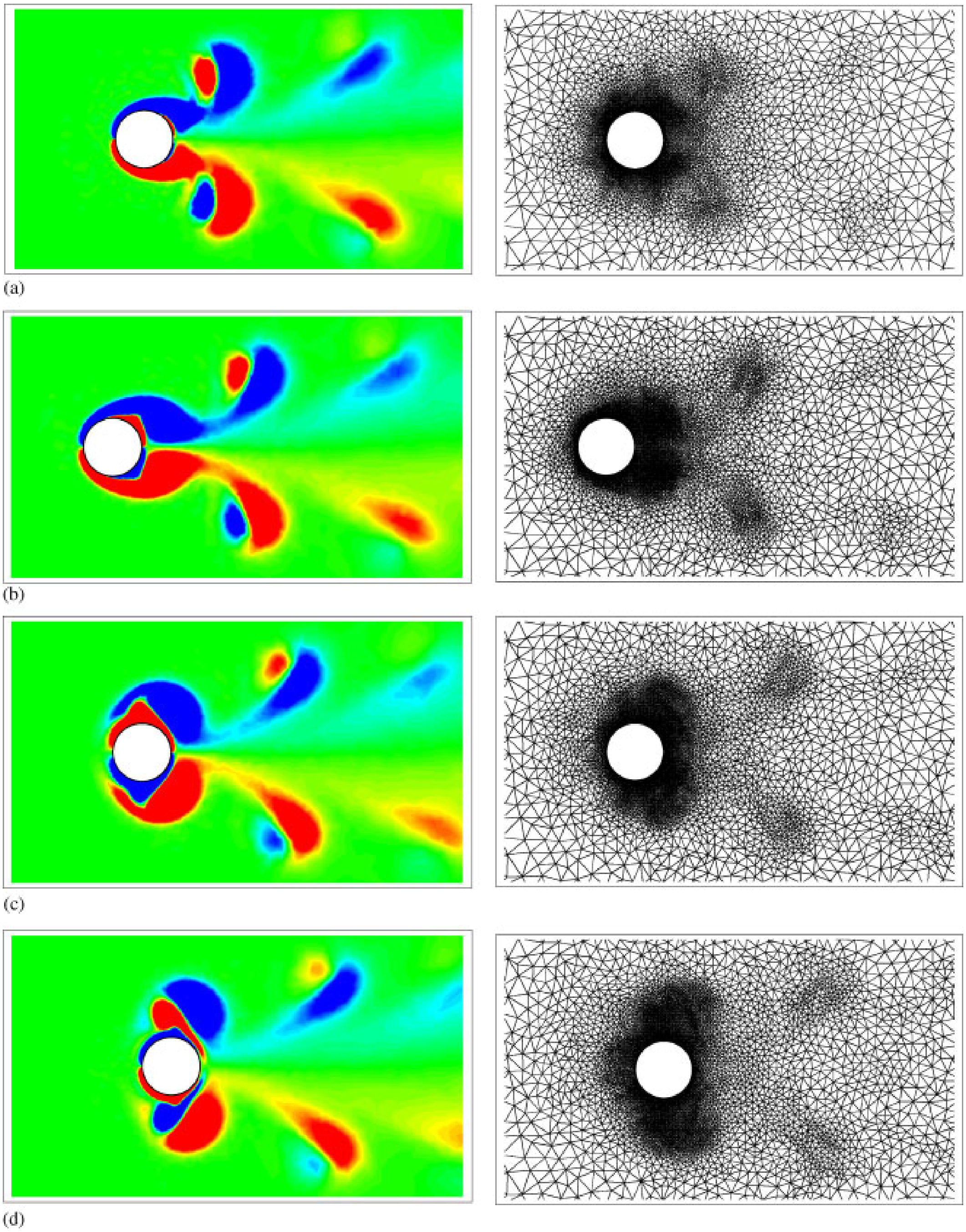

Figure 11. Sequence of frames over one period of oscillation. (a) ω field and adapted grid; (b) ω field and adapted grid, $\frac{1}{4}$ period later; (c) ω field and adapted grid, $\frac{1}{2}$ period later; and (d) ω field and adapted grid, $\frac{3}{4}$ period later.

cylinder location, centre at $x = 0.5$. The second, third, and fourth rows show the location of the cylinder when its centre is at locations $x = 0$, $x = 0.5$, and $x = 1$, respectively. The number of grid points over this period varies between 7500 and 8800. Note that during each period of forced oscillation, two symmetrical pairs of vortices are shed from the cylinder. The pair of vortices that are shed when the cylinder moves in the opposite direction to the flow is stronger than the pair of vortices that are shed when the cylinder moves in the same direction as the flow.

The viscous computations considered in this paper are for flowfields where the Reynolds number is sufficiently small that the flow field is laminar. For this regime it is sufficient to consider a mesh composed of isotropic cells (i.e. cells whose aspect ratio is small). For higher Reynolds number flows, use of isotropic cells leads to a very large number of cells. Although this does not cause any difficulty to the mesh adaptation approach or the flow solver, the CPU time required could render the calculation impractical. For this approach to be practical for engineering simulations on current computers, it would be necessary to introduce an anisotropic mesh with high aspect ratio cells in boundary layers and wake regions. Although the example problems used in this work have fairly simple shapes, these examples were chosen in order to make comparisons with experimental data and with other computations. The dynamic adaptation scheme presented in the paper should apply equally well to any shape no matter how complex.

With the exception of generating the sequence of coarser grids for multigrid by use of an initial triangulation technique, the mesh movement and modification strategy does generalize to 3D. For the 3D case, the sequence of coarser grids for multigrid would be generated by coarsening the fine mesh through edge collapse techniques. A description of the 3D implementation can be found in Reference [37]. It would therefore be possible to extend this work to 3D by coupling the mesh adaptation routines with a 3D Navier–Stokes flow solver.

CONCLUSION

An implicit multigrid-driven algorithm for two-dimensional incompressible laminar viscous flows is coupled with a solution adaptation method to improve the resolution of the flow field and a mesh movement method to handle time dependent boundaries. Validation of the method with unsteady periodic shedding of a circular cylinder in the Reynolds number range of $49 < Re < 180$ shows that the mesh adaptation method is able to track the vortices as they convect downstream and out of the domain and that the Strouhal frequency and base pressure coefficient agree with experimental results and other computational results. Preliminary validation of the time dependent boundary movement method was performed by comparison with experimental results of an oscillating airfoil and calculated results for an oscillating cylinder. These results strengthen one's confidence that this method shows promise for studying more complex configurations.

ACKNOWLEDGEMENTS

The first author wishes to thank Dr Biing-horng Liou for his generous help and Professor C.H.K. Williamson for providing his experimental data. He also wishes to thank Professors Juan Alonso and Brian Helenbrook and Drs Andrey Belov and Geoff Cowles for helpful discussions. The first author is grateful for the generous support of the Princeton University Department of Mechanical and Aerospace

Engineering Guggenheim Foundation for one year and the Office of Naval Research (ONR) Augmentation Award for Science and Engineering Research Training (AASERT) for several years. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

1. Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*, *10th AIAA CFD Conference*, Honolulu, HI, June 1991.
2. Melson ND, Sanetrik MD, Atkins HL. Time-accurate Navier–Stokes calculations with multigrid acceleration. *Proceedings of the 6th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO, 1993.
3. Alonso JJ, Martinelli L, Jameson A. Multigrid unsteady Navier–Stokes calculation with aeroelastic applications. *AIAA Paper 95-0048*, *33rd Aerospace Science Meeting*, Reno, NV, January 1995.
4. Belov A. A new implicit multigrid-driven algorithm for unsteady incompressible flow calculations on parallel computers. *Ph.D. Thesis*, Department of Mechanical and Aerospace Engineering, Princeton University, June 1997.
5. Liou B-H, Martinelli L, Baker TJ, Jameson A. Calculation of plunging breakers with a fully-implicit adaptive-grid method. *AIAA Paper 98-2968*, *29th AIAA Fluid Dynamic Conference*, Albuquerque, NM, 1998.
6. Liou B-H, Balsara D. An implicit unstructured adaptive-grid approach for compressible flows with moving boundaries. *AIAA Paper 2001-0440*, *39th AIAA Aerospace Science Meeting*, Reno, NV, January 2001.
7. Vassberg JC. A fast, implicit unstructured-mesh Euler method. *AIAA Paper 92-2693*, *10th AIAA App. Aerospace Conference*, Palo Alto, CA, June 1992.
8. Lin P. Implicit time dependent calculations for compressible and incompressible flows on unstructured meshes. *Master's Thesis*, Princeton University, Princeton, NJ, November 1994.
9. Venkatakrishnan V, Mavriplis D. Implicit method for the computation of unsteady flows on unstructured grids. *Journal of Computational Physics* 1996; **127**:380–397.
10. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
11. Peyret RJ. Unsteady evolution of a horizontal jet in a stratified fluid. *Journal of Fluid Mechanics* 1976; **78**:49–63.
12. Merkle CL, Athavale M. Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. *AIAA Paper 87-1137*, *8th AIAA CFD Conference*, Honolulu, HI, June 1987.
13. Rogers SE, Kwak D. Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. *AIAA Journal* 1990; **28**(2):253–262.
14. Löhner R. An adaptive finite element scheme for transient problems in CFD. *Computer Methods in Applied Mechanics and Engineering* 1987; **61**:323–338.
15. Baum JD, Löhner R. Numerical simulation of shock interactions with a modern main battlefield tank. *AIAA Paper 91-1666*, *22nd AIAA Fluid Dynamic Conference*, Honolulu, HI, 1991.
16. Connell SD, Holmes DG. A 3D unstructured adaptive multigrid scheme for the Euler equation. *AIAA Paper 93-3339-CP*, *11th AIAA CFD Conference*, Orlando, FL, 1993.
17. Kallinderis Y, Vijayan P. Adaptive refinement-coarsening scheme for three-dimensional unstructured meshes. *AIAA Journal* 1993; **31**(8):1440–1447.
18. Biswas R, Strawn R. A new procedure for dynamic adaption of three-dimensional unstructured grids. *Applied Numerical Mathematics* 1994; **13**:437–452.
19. Delanaye M, Essers JA. An accurate finite volume scheme for Euler and Navier–Stokes equations on unstructured adaptive grids. *AIAA Paper 95-1710-CP*, *12th AIAA CFD Conference*, San Diego, CA, 1995.
20. De Cougny HL, Shephard MS. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering* 1999; **46**:1101–1125.
21. Mavriplis DJ. Turbulent flow calculations using unstructured and adaptive meshes. *International Journal for Numerical Methods in Fluids* 1991; **13**:1131–1152.
22. Mitty TJ, Jameson A, Baker TJ. Solution of three-dimensional supersonic flowflelds via adapting unstructured meshes. *Computers and Fluids* 1993; **22**(2/3):271–283.
23. Marcum DL, Weatherill NP. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal* 1995; **33**(9):1619–1625.
24. Hassan O, Bayne L, Morgan K, Weatherill N. An adaptive unstructured mesh method for transient flows involving moving boundaries. *Technical Report*, ECCOMAS 98, September 1998.
25. Baker TJ. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design* 1997; **25**:243–273.

26. Batina JT. Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA Journal* 1991; **29**(3):327–333.
27. Helenbrook BT. A two-fluid spectral element method. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:273–294.
28. Löhner R. An adaptive finite element solver for transient problems with moving bodies. *Computers and Structures* 1988; **30**(1/2):303–317.
29. Baum JD, Luo H, Löhner R. A new ALE adaptive unstructured methodology for the simulation of moving bodies. *AIAA Paper 94-0414*, *32nd AIAA Aerospace Science Meeting*, Reno, NV, 1994.
30. Baker TJ, Cavallo PA. Dynamic adaptation for deforming tetrahedral meshes. *AIAA Paper 99-3253*, *14th AIAA CFD Conference*, Reston, VA, 1999.
31. Cavallo PA, Lee RA, Hosangadi A, Dash SM. Dynamic unstructured grid methodology with applications to aero/propulsive flowfields. *AIAA Paper 97-2310*, 1997.
32. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**: 73–94.
33. Mittal S, Tezduyar TE. A finite element study of incompressible flows past oscillating cylinders and aerofoils. *International Journal for Numerical Methods in Fluids* 1991; **29**(3):327–333.
34. Rausch RD, Batina J, Yang H. Spatial adaptation procedures on tetrahedral meshes for unsteady aerodynamic flow calculations. *AIAA Paper 93-0670*, *31st AIAA Aerospace Science Meeting*, Reno, NV, 1993.
35. Johnson AA, Tezduyar TE. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**:351–373.
36. Baker TJ. Mesh modification for solution adaptation and time evolving domains. *7th International Conference on Numerical Grid General in Computer Field Simulations*, Whistler, BC, Canada, 2000.
37. Baker TJ. Mesh movement and metamorphosis. *Engineering with Computers* 2002; **18**(3):188–198.
38. Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer–Watson algorithm. *Journal of Computational Physics* 1993; **106**:125–138.
39. Mavriplis DJ, Jameson A, Martinelli L. Multigrid solution of the Navier–Stokes equations on triangular meshes. *AIAA Paper 89-0120*, *27th AIAA Aerospace Science Meeting*, Reno, NV, 1989.
40. Barth TJ. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. *AGARD Report 787 Special Course on Unstructured Grid Methods for Advection Dominated Flows*, 1992; 6-1–6-61.
41. Rizzi A, Eriksson L-E. Computation of inviscid incompressible flow with rotation. *Journal of Fluid Mechanics* 1985; **153**:275–312.
42. Dreyer JJ. Finite volume solutions to the steady incompressible Euler equations on unstructured triangular meshes. *Master's Thesis*, Princeton University, Princeton, NJ, 1990.
43. Mavriplis DJ. Multigrid solution of the 2-D Euler equations on unstructured triangular meshes. *AIAA Journal* 1988; **26**(7):824–831.
44. Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers* 1989; **5**:161–175.
45. Bowyer A. Computing Dirichlet tessellations. *Computer Journal* 1981; **24**(2):162–166.
46. de Sampaio PAB, Lyra PRM, Morgan K, Weatherill NP. Petrov–Galerkin solutions of the incompressible Navier–Stokes equations in primitive variables with adaptive remeshing. *CMAME* 1993; **106**:143–178.
47. Williamson CHK. Defining a universal and continuous Strouhal–Reynolds number relationship of the laminar vortex shedding of a circular cylinder. *Physics of Fluids* 1988; **31**:2742–2744.
48. Williamson CHK, Brown G. A series in $1/\sqrt{Re}$ to represent the Strouhal–Reynolds number relationship of the cylinder wake. *Journal of Fluids and Structures* 1998; **12**:1073–1085.
49. Williamson CHK, Roshko A. Measurements of base pressure in the wake of a cylinder at low Reynolds numbers. *Zeitschrift fur Flugwissenschaften und Weltraumforschung* 1990; **14**:38–46.
50. Henderson RD. Unstructured spectral element methods: parallel algorithms and simulations. *Ph.D. Thesis*, Department of Mechanical and Aerospace Engineering, Princeton University, 1994.
51. Halfman RL. Report no. 1108 experimental aerodynamic derivatives of a sinusoidally oscillating airfoil in two-dimensional flow. *Thirty-Eight Annual Report of the NACA 1952 Including Technical Report Nos. 1059-1110*, 1954.
52. Sarpkaya T. Vortex-induced oscillations. *Journal of Applied Mechanics* 1979; **46**:241–258.