AIAA JOURNAL
Vol. 37, No. 12, December 1999

Flow Simulation Using Generalized Static and Dynamic Grids

R. P. Koomullil* and B. K. Soni†
Mississippi State University, Mississippi State, Mississippi 39762

A new approach is presented for a flow simulation system using generalized grids. In a generalized grid, the physical domain of interest is decomposed into cells with an arbitrary number of edges or faces. The grid can be of structured, unstructured, or hanging node type or an arbitrary combination of the types. A cell-face-based data structure is used to store the grid information. A flow simulation system is developed for generalized grids that can handle static and dynamic grids. The full Navier–Stokes equations, in the integral form, are taken as the relations that govern the fluid flow. A cell-centered finite volume scheme is developed for solving these governing equations. The numerical flux across the cell faces is evaluated by an upwind scheme based on Roe's approximate Riemann solver. A higher-order scheme is formulated by utilizing Taylor's series expansion and Green's theorem. Limiter functions are used to preserve monotonocity. The skin-friction coefficient is used to evaluate the accuracy of the limiter functions. The generalized minimal residual method is utilized to solve the sparse linear system of equations resulting from the linearization of the flux vectors. The Spalart–Allmaras one-equation turbulence model has been implemented for the generalized grids and is used to evaluate the turbulent viscosity. For dynamically moving bodies, the equations of classical mechanics are used to predict the trajectory based on the external aerodynamic and body forces. A variety of computational examples are presented to demonstrate the wide range of applications, and the results are compared with experimental data whenever available.

I. Introduction

DURING the past decade, computational simulation of fluid flow over complex configurations has progressed significantly, and numerous notable successes have been reported in the literature[1–7]; however, the generation of a high-quality mesh for such problems has often been reported as a pacing item. Hence, much effort has been expended to speed up this portion of the simulation process. Two of the most common approaches to grid generation are structured multiblock[8,9] and unstructured,[10–13] primarily tetrahedron-based, procedures.

Flow solvers based on structured grids tend to be computationally more efficient than those based on unstructured grids. High-aspect-ratio cells necessary for the resolution of viscous boundary layers can be easily generated and can be used with existing structured grid-based flow solvers.[7,14] In the case of structured grids for complex configurations, the physical domain has to be decomposed into different subdomains (blocks), and the grid has to be generated separately for individual blocks. In many cases, grid lines do not exhibit continuity across the block interfaces. Even with this relaxation of the continuity of the grid lines, the grid-generation process is time consuming. Flow solvers supporting noncontiguous interfaces require specialized interpolation procedures to preserve the conservation of the flow variables.

Flow solvers based on an unstructured mesh require more CPU time and memory per grid point because the number of grid points necessary to resolve the boundary layers using nearly equilateral triangles is enormous, resulting in significantly higher CPU and memory requirements. Recently, success has been reported in generating high-aspect-ratio unstructured grids for viscous simulations using an advancing normal point placement strategy.[15,16] In this case, the triangles inside the boundary layer are very skewed and will affect the accuracy of the solution.

The generalized data structure of an unstructured grid is very useful in refinement or derefinement. The addition or removal of the grid points affects only the local data structure. Adaptation of the grid to the flow features[17,18] is then easier compared with moving or adding grid points to a structure grid.

Hybrid or generalized grid-generation and solution techniques have been developed with the objective of combining the attractive features of both structured and unstructured techniques. Nakahashi et al.[19] used a zonal method for the hybrid grid generation. In their approach, a body-fitted structured grid is kept in the boundary layer, and a finite difference scheme is used to solve the Reynolds-averaged Navier–Stokes equations in the structured portion of the grid. The rest of the domain is filled with tetrahedrons, and the Euler equations are solved in that region. The interface between the structured and unstructured regions is treated as an explicit boundary. Another approach, by Lohner,[20] used a combination of semistructured and unstructured grids for getting high-aspect-ratio cells in the boundary layer. The semistructured grid in the boundary layer is generated using the surface normals. In this approach, prisms are generated in the boundary layer and are trimmed to avoid grid crossing. These prisms are subdivided into tetrahedrons, resulting in only tetrahedrons in the domain. The main disadvantage of this method is the presence of highly skewed tetrahedrons in the boundary layer. The approach by Sharov and Nakahashi[21] and Kallinderis et al.[22] used semistructured prisms in the viscous regions and tetrahedra in the rest of the domain. In both of these cases, the marching direction was used as the surface normals to generate the prisms. In the approach by Weatherill[23] and Kao and Liou,[24] most of the domain is filled with structured elements, and the different components of the structured grid are connected using unstructured grids. This technique involves the decomposition of complex geometries into a number of simple geometric entities. Structured grids are then generated around the simplified entities. One of the structured grids is termed the main grid. All of the remaining component grids are contained within it. The main grid and the component grids are overlaid, and holes are cut in the main grid where the structured component grid has to be placed. The voids between the main and the component grids are filled with an unstructured grid. In this approach, the transition across structured and unstructured grids may not be smooth due to sudden change in the area ratios of the cells.

In the present hybrid grid-generation approach,[25] structured grids are used only near solid bodies, and the rest of the domain is filled with unstructured grids. The structured grid generation is based on an advancing-layer-type method, and the unstructured grid is based on Delaunay triangulation. The quality of the cells at the interfaces is ensured by checking the aspect ratios of the cell. In the present flow solver, the structured and unstructured grids are treated as a single block through a generalized data structure that facilitates the handling of the cells with any arbitrary number of sides. For two-dimensional cases, an edge-based data structure is used for

Received 19 November 1997; revision received 4 February 1999; accepted for publication 30 April 1999. Copyright © 1999 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Visiting Research Assistant, P.O. Box 9627, Engineering Research Center. Member AIAA.

†Professor, P.O. Box 9627, Department of Aerospace Engineering, Engineering Research Center. Senior Member AIAA.

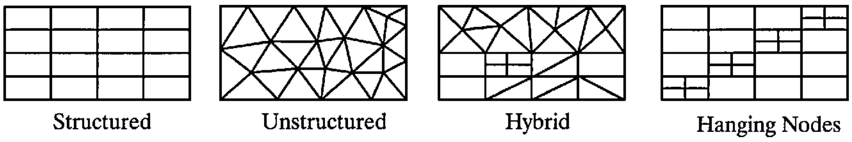


Fig. 1 Generalized grids.

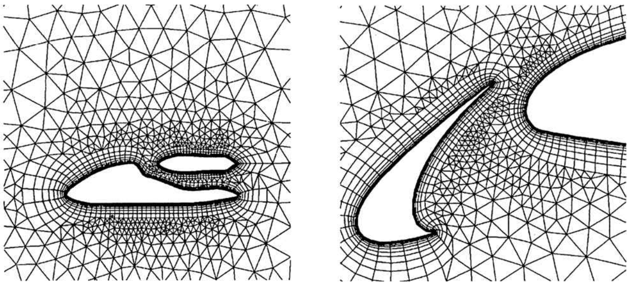


Fig. 2 Examples of hybrid grid.

this purpose, and a cell-face-based data structure is used for three-dimensional cases. In the edge-based data structure, every edge that forms the grid has four pieces of information associated with it: the nodes that form the edge and the cell numbers on either side of the edge. For the boundary faces, the boundary condition information is stored in the place of one cell number. In three-dimensional cases, for each cell face the number of nodes that form the face, the node numbers, and the cell on both sides are stored. The generalized data structure easily accommodates the grid types demonstrated in Fig. 1. Two examples of hybrid grids are given in Fig. 2.

Early development associated with the flow solvers for hybrid grids, structured and unstructured grids, are treated as two different blocks,[26,27] and explicit boundary conditions are used to transfer information between them. This introduces a time lag between the interface of the structured and the unstructured grids, resulting in a possible convergence degradation. Two other approaches in the field of hybrid flow solvers are due to Weatherill[23] and Parthasarathy et al.[28] The approach by Weatherill[23] utilizes a multistage time-stepping scheme together with artificial dissipation. In the other approach, Parthasarathy et al.[28] use a Lax–Wendroff temporal discretization.

The full Navier–Stokes equations, in the integral form, are taken as the governing equations for the flow simulation. A cell-centered, finite volume, upwind scheme based on Roe's approximate Riemann solver[29] is used for solving these governing equations. The higher-order accuracy is achieved using Taylor's series expansion and Green's theorem. The monotonocity of the conserved variables is preserved using a limiter function. The generalized minimal residual (GMRES)[30] method is utilized to solve the sparse linear system of equations resulting from the linearization of the flux vectors. The flux Jacobians are evaluated using either an approximate analytical method or a numerical differentiation procedure. The boundary conditions based on the characteristic variables are implemented for generalized grids. The viscous fluxes are evaluated explicitly. The Spalart–Allmaras one-equation turbulence model[31] has been extended to accommodate the generalized grid and is used to evaluate the turbulent viscosity. For dynamically moving bodies, the equations of classical mechanics are used to predict the trajectory based on the external aerodynamic and body forces.

The details about the algorithms related to the flow solver are given in Sec. II. The validation of the code and the capabilities of the present approach to handle moving body problems are presented in Sec. III. Section III also contains the results from the algorithmic studies. Conclusions from the present study are summarized in Sec. IV.

II. Flow Solver Development

The nondimensionalized, integral form of the Navier–Stokes equations, with standard notations for conserved variables Q, the convective flux vector $\boldsymbol{F}$, and the viscous flux vector $\boldsymbol{F}^{v}$, applied to a control volume Ω that is bounded by a control surface $\partial\Omega$ can be written as

$$\frac{\partial}{\partial t}\int_{\Omega} Q\,\mathrm{d}\Omega + \oint_{\partial\Omega} \boldsymbol{F}(Q)\cdot\boldsymbol{n}\,\mathrm{d}s = \oint_{\partial\Omega} \boldsymbol{F}^{v}(Q)\cdot\boldsymbol{n}\,\mathrm{d}s \tag{1}$$

where n is the outward pointing surface normal and $\mathrm{d}s$ is the cell-face area. The nondimensionalizations are done using the freestream conditions. The velocity components are nondimensionalized with respect to the total freestream velocity.

Finite volume schemes are best suited for hybrid grids because a typical hybrid grid is an agglomeration of polygons with different numbers of sides per polygon. For the present work we have used a cell-centered finite volume scheme in which cell-averaged flow variables are stored at the cell center. A semidiscretized form of Eq. (1), without body forces, is written as

$$\frac{\partial}{\partial t}\int_{\Omega} Q\,\mathrm{d}\Omega = -\sum_{j=1}^{k} \boldsymbol{F}_{ij}\cdot\boldsymbol{n}_j\,\mathrm{d}s_j \tag{2}$$

where i and j denote cell and face numbers, respectively, and k is the total number of faces of the ith cell. The numerical flux passing through the cell faces is calculated by Roe's approximate Riemann solver[29] and is given by

$$F_{ij} = \tfrac{1}{2}[F(Q_R) + F(Q_L) - |\bar{A}|(Q_R - Q_L)] \tag{3}$$

where R and L represent the cells on the right- and the left-hand sides of the cell face, respectively, and $|\bar{A}| = T|\Lambda|T^{-1}$, where T is the matrix whose columns are the right eigenvectors of $\bar{A}$, T^{-1} is the matrix whose rows are the left eigenvectors of $\bar{A}$, and $|\Lambda|$ is a diagonal matrix whose elements are the absolute values of the eigenvalues of $\bar{A}$. The matrix $\bar{A}$ is evaluated using the Roe-averaged variables.[29]

Higher-order accuracy in the spatial discretization is obtained by using a linear reconstruction of the conserved variables that is obtained using Taylor's series expansion and Gauss's theorem. During the reconstruction process, local extrema that may be created are avoided by using limiter functions. The Taylor series expansion with limiter function ϕ_l is written as

$$Q(x, y, z) = Q(x_i, y_i, z_i) + \phi_i \nabla Q(x_i, y_i, z_i)\cdot\Delta\boldsymbol{r} + \mathcal{O}[(\Delta\boldsymbol{r})^2] \tag{4}$$

where ∇Q is the gradient of Q and $\nabla\boldsymbol{r}$ is the vector from the cell center (x_i, y_i, z_i) to the desired point (x, y, z).

The gradient of Q at the cell center is estimated using Green's theorem, with the control volume taken as the cell itself. The flow variables at the cell center are calculated using a weighted-averaging approach. Two different limiter functions were considered, and their effect on the skin friction was studied.

A. Barth and Jespersen Limiter

A widely used limiter for unstructured grids was developed by Barth and Jespersen[32] and is defined as

$$\phi_{ij} = \begin{cases} \min\left(1, \dfrac{Q_i^{\max} - Q_{ij}}{Q_i - Q_{ij}}\right) & \text{if} \quad Q_i - Q_{ij} > 0 \\ \min\left(1, \dfrac{Q_i^{\min} - Q_{ij}}{Q_i - Q_{ij}}\right) & \text{if} \quad Q_i - Q_{ij} < 0 \\ 1 & \text{if} \quad Q_i - Q_{ij} = 0 \end{cases} \tag{5}$$

where Q_{ij} is calculated using Eq. (4) without the limiter function ϕ_i.

B. Venkatakrishnan's Limiter

Venkatakrishnan[33] proposed another limiter for unstructured grids that is an extension of the van Albada limiter that was developed for structured grids. This limiter is defined as

$$\phi_{ij} = \frac{1}{\Delta_-}\left[\frac{\left(\Delta_+^2 + \epsilon^2\right)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_-\,\Delta_+ + \epsilon^2}\right] \tag{6}$$

where ϵ^2 is taken as $(K\Delta x)^3$ and Δx is the average edge size of triangles; that is, if the physical domain is covered with the same number of equilateral triangles of equal size, then Δx is the length of the edge of those triangles and K is a threshold parameter. When $K = 0$, the limiter will be active everywhere in the field, whereas a very high value of K implies effectively no limiting.

C. Implicit Scheme and Newton Iterations

In the case of implicit schemes, the numerical flux crossing the cell face is a function of the conserved variables at the $(n+1)$th time level. The flux vector has to be linearized before the evaluation of the flux crossing the cell faces. For the moving grids, the cell volume is changing at every time step and has to satisfy the geometric conservation law. For unsteady calculations, it is important to drive the unsteady residual to zero for better resolution of the physical phenomenon. This process is accomplished by using Newton iterations.[34] After linearization and simplification, the resulting linear system will reduce to

$$\begin{aligned}&\left[\frac{V_i}{\Delta t}I + \sum_{j=1}^{k}\left(\frac{\partial F_{ij}}{\partial Q_i}\right)^{n+1,m}\right]\Delta Q_i^{n+1,m}\\&\quad + \sum_{j=1}^{k}\left(\frac{\partial F_{ij}}{\partial Q_j}\right)^{n+1,m}\Delta Q_j^{n+1,m}\\&\quad = -\left[V_i\left(\frac{Q_i^{n+1,m} - Q_i^n}{\Delta t}\right) + \sum_{j=1}^{k}F_{ij}(Q^{n+1,m})\right]\end{aligned} \tag{7}$$

where j are the cells that surround the cell i and the superscript m is the index for Newton iterations. At the start of the Newton iteration, $Q^{n+1,0}$ is set as Q^n. The sparse matrix system resulting from Eq. (7) is solved using GMRES.[30]

D. Approximate and Numerical Jacobians

The Roe-averaged matrix $|\bar{A}|$ is a nonlinear function of the conserved variables at the left- and the right-hand sides of the cell face. For simplicity, the approximate analytic Jacobians are calculated assuming the Roe-averaged matrix $|\bar{A}|$ as constant. The relation for numerical flux [Eq. (3)] is then differentiated analytically with respect to Q_R and Q_L.

The numerical Jacobian is calculated from the first principles. Each element in the Jacobian is written as $a_{ij} = [\partial F_i(Q)/\partial Q_j]$ and is evaluated using the finite difference formula[35]

$$a_{ij} = \frac{F_i(Q + h\boldsymbol{e}_j) - F_i(Q)}{h} \tag{8}$$

where $\boldsymbol{e}_j$ is the jth unit vector and h is taken as the square root of machine zero.

E. Turbulence Modeling

The simulation of many complex features of the flows of practical importance needs to account for the flow's turbulent behavior. The laminar viscosity is usually a function of temperature and is estimated using Sutherland's formula.[36] The turbulent viscosity is a function of the flow and usually evaluated by using an empirical model. In this study the turbulent viscosity is estimated using the Spalart–Allmaras one-equation model[31] and the Reynolds stress is modeled using the Boussinesq hypothesis.[37]

The Spalart–Allmaras one-equation model encompasses a solution of a second-order partial differential equation for the variable $\bar{\nu}$, and the turbulent kinematic viscosity ν_t is estimated from $\bar{\nu}$ by applying a damping function f_{v1}. The nondimensional, integral form of the associated governing equation, without tripping terms, is given by

$$\begin{aligned}\int_\Omega \frac{\partial\bar{\nu}}{\partial t}\,\mathrm{d}\Omega &= -\int_\Omega \boldsymbol{V}\cdot\nabla\bar{\nu}\,\mathrm{d}\Omega + \int_\Omega C_{b1}\tilde{S}\nu\,\mathrm{d}\Omega\\&\quad + \frac{(1 + C_2)}{Re_L\sigma}\oint_{\partial\Omega}(\nu + \bar{\nu})\nabla\bar{\nu}\cdot\boldsymbol{n}\,\mathrm{d}s - \frac{C_{b2}}{Re_L\sigma}\int_\Omega(\nu + \bar{\nu})\nabla^2\bar{\nu}\,\mathrm{d}\Omega\\&\quad - \frac{C_{\omega1}}{Re_L}\int_\Omega f_\omega\left(\frac{\bar{\nu}}{d}\right)^2\mathrm{d}\Omega\end{aligned} \tag{9}$$

The details of the model are summarized as

$$C_{b1} = 0.1355, \qquad \sigma = \frac{2}{3}, \qquad C_{b2} = 0.622$$

$$\kappa = 0.41, \qquad C_{\omega1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma}, \qquad C_{\omega2} = 0.3$$

$$C_{\omega3} = 2.0, \qquad C_{v1} = 7.1, \qquad \chi = \frac{\bar{\nu}}{\nu}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \qquad \nu_t = \bar{\nu}f_{v1}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

$$\tilde{S} = S + \frac{1}{Re_L}\left(\frac{\bar{\nu}}{\kappa^2 d^2}\right)f_{v2}, \qquad f_\omega = g\left(\frac{1 + C_{\omega3}^6}{g^6 + C_{\omega3}^6}\right)^{\frac{1}{6}}$$

$$g = r + C_{\omega2}(r^6 - r), \qquad r = \frac{1}{Re_L}\left(\frac{\bar{\nu}}{\tilde{S}\kappa^2 d^2}\right)$$

The discretized form of Eq. (9) is written as

$$\begin{aligned}\frac{\bar{\nu}^{n+1} - \bar{\nu}^n}{\Delta t_i}V_i &= -\sum_{j=1}^{k}\left(U^+\bar{\nu}_i + U^-\bar{\nu}_{n(j)}\right)\mathrm{d}s_j + C_{b1}(\tilde{S}\bar{\nu})_i V_i\\&\quad + \frac{(1 + C_{b2})}{\sigma Re_L}\sum_{j=1}^{k}(\nu+\bar{\nu})_{ij}(\nabla\bar{\nu}\cdot\boldsymbol{n})_{ij}\,\mathrm{d}s_j - \frac{C_{b2}}{\sigma Re_L}(\nu+\bar{\nu})_i\nabla^2\bar{\nu}_i V_i\\&\quad - \frac{C_{\omega1}}{Re_L}\left[f_\omega\left(\frac{\bar{\nu}}{d}\right)^2\right]_i V_i\end{aligned} \tag{10}$$

where k is the number of edges of cell i, $n_{(j)}$ is the cell that shares the jth edge of the cell i, and V_i is the cell volume. The variables U^+ and U^- are defined as

$$U^+ = \tfrac{1}{2}(U + |U|), \qquad U^- = \tfrac{1}{2}(U - |U|)$$

where U is the contravariant velocity. For the explicit scheme, a four-stage Runge–Kutta method is applied. For implicit calculations the Jacobians are calculated numerically,[38] and the resulting sparse matrix is solved using GMRES.[30]

F. Unsteady and Dynamic Motion Calculations

The geometric conservation law adds a correction term to the governing equations to account for the grid motion.[39] This correction preserves the uniform conditions when the grid moves, thereby avoiding the creation of spurious sources and sinks in the flowfield. The geometrical conservation law in integral form, which relates the rate of change of the control volume to the orientation and velocities $\boldsymbol{c}$ of the cell faces, is written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \mathrm{d}\Omega = \oint_{\partial\Omega}\boldsymbol{c}\cdot\boldsymbol{n}\,\mathrm{d}s \tag{11}$$

For flow simulation over moving bodies, the grid has to be regenerated globally at each time step, or the grid points have to be moved appropriately to retain a body-conforming grid without overlapping cells. Regeneration of grids for each time step is expensive, and interpolation has to be done to transfer the data between the grids. The interpolation is usually nonconservative and reduces the accuracy of the simulation. In the present study, the tension spring analogy is used for grid point movement.[40]

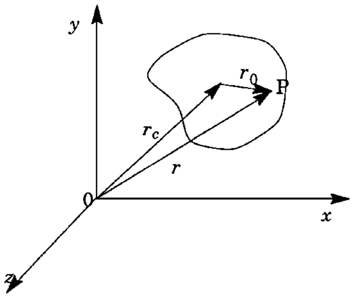


Fig. 3 Notation used for rigid-body dynamics.

In the tension spring analogy, each edge of the grid is assumed to be a tension spring. The movement of the interior points is calculated by solving a system of tension springs when the boundary points are disturbed. The spring analogy, by specifying the boundary displacement, will result in a linear system for the interior point displacements. This system is written as

$$\sum_j K_{ij}(\Delta x_i - \Delta x_j) = 0, \qquad \sum_j K_{ij}(\Delta y_i - \Delta y_j) = 0 \tag{12}$$

where K_{ij} is the spring stiffness corresponding to the edge connecting nodes i and j and is defined as l_{ij}^{-2}, where l_{ij} is the length of the edge. The summation varies over all nodes that are connected to the node i. The sparse matrix system resulting from Eqs. (12) is solved using GMRES. For the situations involving large relative motion of the bodies, the quality of the grid will degrade after a few hundred time steps. At this stage, a new grid is generated, and the solution is interpolated to the new grid.

The trajectory of moving bodies is determined by the laws of classical mechanics. It is assumed that all moving bodies have six degrees of freedom. These six degrees of freedom are the translation along the three coordinate axes and the rotation about the three coordinate axes. The movement of the body is determined based on the pressure distribution over the body and the gravitational force due to the weight of the body.[1]

The notations used for rigid-body motion are shown in Fig. 3. The position vector of any point on the surface of the body is expressed as the vector sum of the position vector of the center of gravity and the vector from the center of gravity to the corresponding point and is written as

$$\boldsymbol{r} = \boldsymbol{r}_c + \boldsymbol{r}_0 \tag{13}$$

Now the velocity at the point P is given by

$$\dot{\boldsymbol{r}} = \dot{\boldsymbol{r}}_c + \dot{\boldsymbol{r}}_0 = \boldsymbol{V}_c + \boldsymbol{\omega} \times \boldsymbol{r}_0 \tag{14}$$

where $\boldsymbol{r}$ are the position vectors of the point P, shown in Fig. 3. The linear and the angular velocities of the body are denoted by $\boldsymbol{V}_c$ and $\boldsymbol{\omega}$, respectively.

The balance of forces and moments acting on the body results in the relations for the translational and rotational accelerations. These expressions are written as

$$m\dot{\boldsymbol{V}}_c = \sum \boldsymbol{F} = m\boldsymbol{g} - \oint_{\partial\Omega} p\boldsymbol{n}\,\mathrm{d}s \tag{15}$$

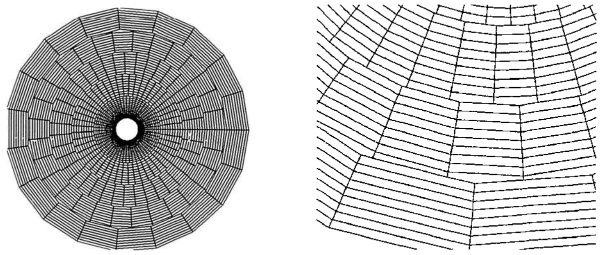
$$\Theta \cdot \dot{\boldsymbol{\omega}} + \int_\Omega (\boldsymbol{\omega} \cdot \boldsymbol{r}_0)(\boldsymbol{r}_0 \times \boldsymbol{\omega})\,\mathrm{d}\Omega = \sum \boldsymbol{r}_0 \times \boldsymbol{F} = -\oint_{\partial\Omega} p\boldsymbol{r}_0 \times \boldsymbol{n}\,\mathrm{d}s \tag{16}$$

where

$$\Theta = \begin{bmatrix} I_{yy} + I_{zz} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{xx} + I_{zz} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{xx} + I_{yy} \end{bmatrix}, \qquad I_{ij} = \int_\Omega r_0^i r_0^j \rho\,\mathrm{d}\Omega$$

III. Results and Discussion

The present approach for the flow simulation has been extensively tested and validated and is presented by Koomullil.[38] The capability of the flow solver to handle the cells with an arbitrary number of sides is demonstrated using the grid shown in Fig. 4, which contains cells with four, five, and six sides. This grid consists of 2345 nodes and 2536 cells. Some of the cells have hanging nodes, where only two edges are connected to that node. The flow conditions taken were a freestream Mach number of 0.30 and an angle of attack of 0 deg. The pressure contours for this simulation along with the grid are shown in Fig. 5.



Overall view of the grid **Zoomed view at the interface between layers**

Fig. 4 Generalized grid containing more than four nodes per cell.

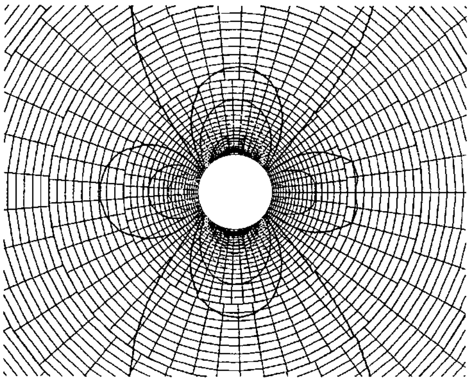


Fig. 5 Contour plot of pressure distribution around circular cylinder at freestream Mach number 0.3.

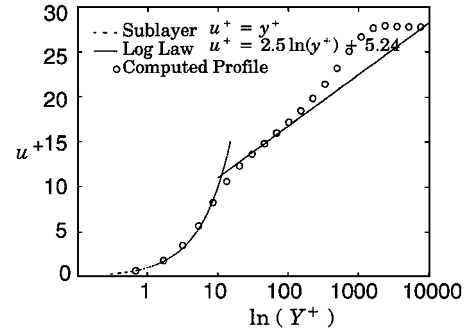


Fig. 6 Turbulent velocity profile compared with theoretical data.

The turbulent flow simulation is validated using the flow over a flat plate. The grid consists of 2911 nodes and 2800 rectangles. The physical domain includes five plate lengths upstream of the leading edge, four plate lengths downstream of the trailing edge, and five plate lengths above the plate. The first point off the plate is at a distance of 1.0×10^{-5}. The freestream Mach number for this simulation is taken as 0.5 and the Reynolds number as 2×10^6. The computed turbulent velocity profile is compared to theoretical results in Fig. 6.

The turbulent flow calculations were carried out over a NACA 0012 airfoil. The flow conditions used were a freestream Mach number of 0.799, an angle of attack of 2.26 deg, and a Reynolds number of 9.0×10^6. The grid used for this simulation is a structured grid of dimension 290×81 (Fig. 7) and is converted into the hybrid grid data structure. A total of 200 points were given on the surface of the airfoil. The Mach number contours are shown in Fig. 8. The explicit scheme takes 1.5×10^{-4} s per iteration per cell, whereas the implicit scheme takes 1.44×10^{-3} s per iteration per cell on a Silicon Graphics (SGI) R8000 processor. For double-precision calculations, the explicit scheme requires 102 words per cell, and the implicit scheme requires 550 words per cell.

A. Effect of Limiter Function on Skin Friction

The effect of the limiter function on the skin-friction coefficient is studied using the flow over the NACA 0012 airfoil. The skin-friction distribution over the airfoil is plotted in Fig. 9. The skin-friction distribution does not change with the different threshold parameters for Venkatakrishnan's limiter.[33] However, Barth and Jespersen's limiter gives higher skin friction at the leading edge of the airfoil and lower skin friction after the leading edge. After the shock-induced separation at the upper surface of the airfoil, both limiters gave the same skin friction except for the threshold parameter $K = 20$.

The computed skin-friction distribution over the flat plate for a Reynolds number of 2×10^6 and a freestream Mach number of 0.5 is compared with the theoretical values in Fig. 10. Venkatakrishnan's limiter[33] underpredicts the skin friction, but Barth and Jespersen's limiter gives a nonsmooth skin-friction coefficient. For very high values of the threshold parameter ($K = 50$), the distribution of the skin friction deviates from the theoretical ones.[37]

B. Unsteady Calculations

The unsteady calculations were validated using inviscid transonic flow over the NACA 0012 airfoil pitching about the quarter chord. A

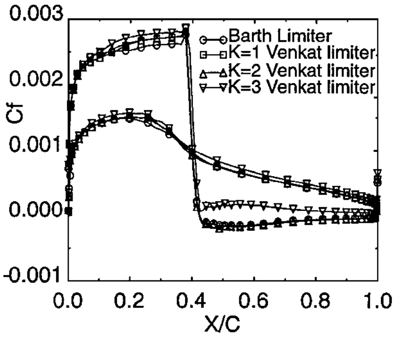


Fig. 9 Skin-friction coefficient distribution over NACA 0012 airfoil.

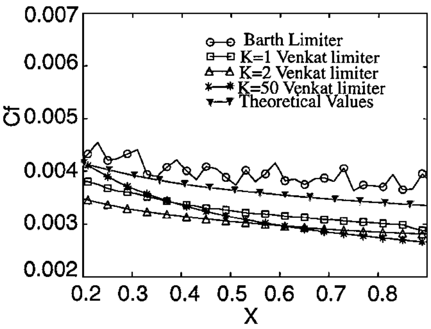


Fig. 10 Skin-friction distribution over a flat plate.

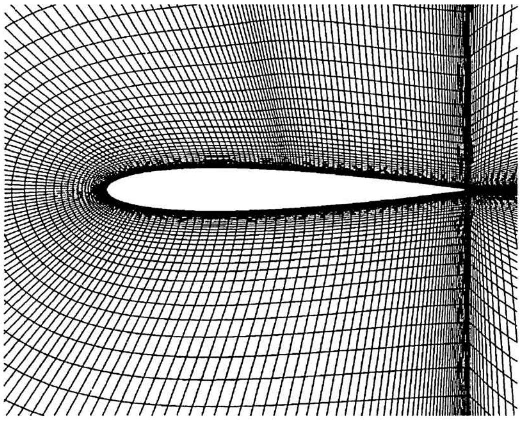
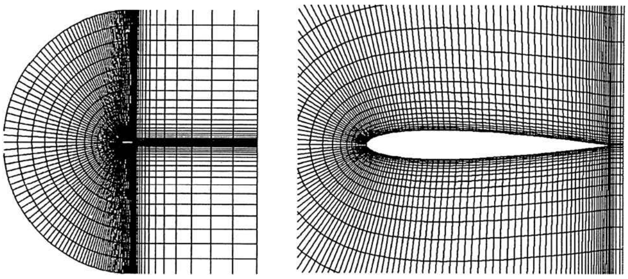


Fig. 7 Grid used for turbulent flow over NACA 0012 airfoil.



Complete grid

Closeup view of the grid

Fig. 11 Grid used for pitching airfoil.

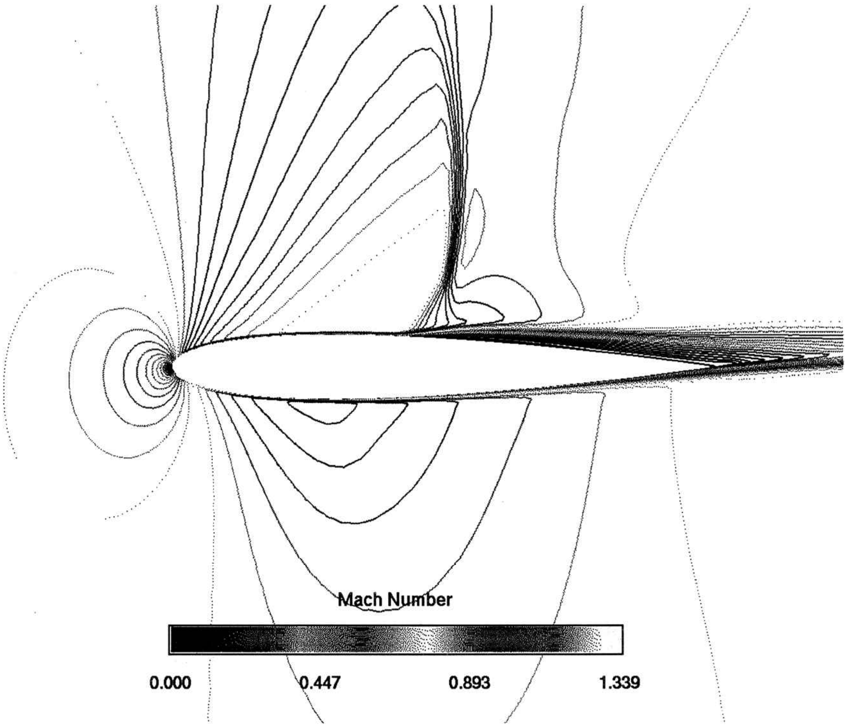


Fig. 8 Mach number contours.

structured grid (shown in Fig. 11) with 6000 cells and 6211 nodes is used for this simulation. The unsteady calculations were started from a steady-state solution. The movement of the airfoil is prescribed such that the angle of attack varies sinusoidally according to the following relation:

$$a(t) = a_m + a_0 \sin(\omega t)$$

where a_m is the mean angle of attack and is taken as 0.016 deg and the maximum deflection a_0 is taken as 2.51 deg. The NACA 0012 airfoil is assumed to be pitching at a reduced frequency k of 0.1628 at a freestream Mach number of 0.755. The reduced frequency is defined as

$$k = \omega c / V_\infty$$

where ω is the frequency in radians per second, c is the chord length, and V_∞ is the freestream velocity. The computed lift history is plotted in Fig. 12. The results are compared with the experimental data.[41] The lift history attains a periodic nature after half a cycle of oscillation. The simulation of a full cycle of oscillation, on an SGI R4400 processor, takes 50 min of CPU time.

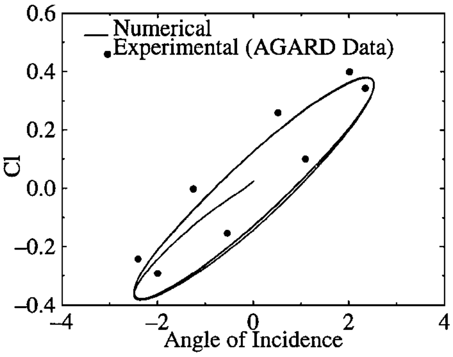
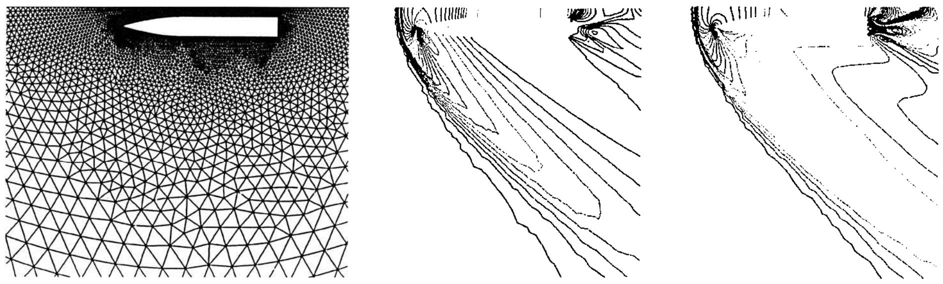


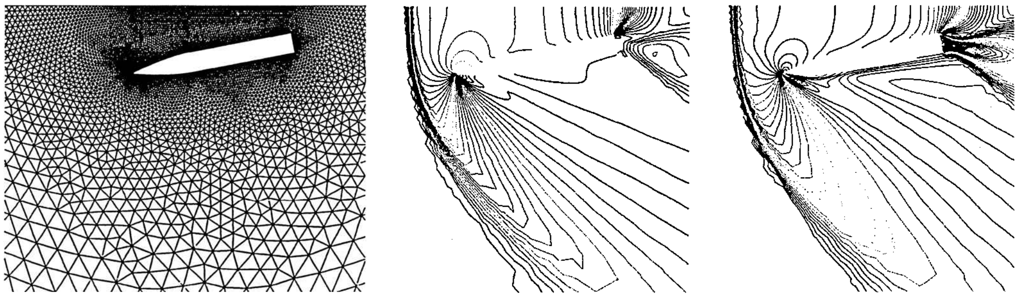
Fig. 12 Lift history for pitching NACA 0012 airfoil.

The capability of the flow simulations involving bodies in relative motion is demonstrated using a simplified model of the store separation problem. The grid used for this simulation consists of 4824 nodes and 9216 cells. The freestream Mach number is taken as 2.0. The pressure and Mach number distributions for the steady-state condition are shown in Fig. 13. The trajectory of the missile is calculated by using the aerodynamic forces acting on it and its weight. During each time iteration, the position of the body changes and the corresponding changes in the interior grid points are calculated using the spring analogy. The grids are regenerated when the cells get too skewed, and the solution is interpolated to the new grid. Regeneration was carried out three different times during the solution process. The grid, pressure distribution, and Mach number distribution at two different time levels are shown in Figs. 14 and 15.



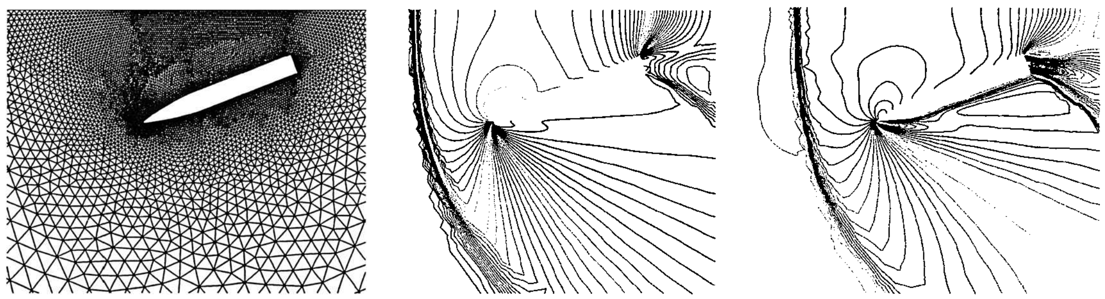
Grid **Pressure** **Mach number**

Fig. 13 Steady-state solution.



Grid **Pressure** **Mach number**

Fig. 14 Solution at $t = 177.52$.



Grid **Pressure** **Mach number**

Fig. 15 Solution at $t = 243.5836$.

IV. Conclusions

A new method has been developed for flow simulation on generalized grids. The results of the computations have been validated with experimental data whenever available. The capability of the code to handle an arbitrary number of cell sides has also been demonstrated. A fully implicit scheme with approximate analytic and numerical Jacobians has been implemented for generalized grids. The implicit scheme with numerical Jacobians takes approximately 5% more CPU time per time iteration than with approximate analytical Jacobians, but a better convergence is achieved. Two different limiter functions have been implemented. For the limiter by Venkatakrishnan,[33] high values of the threshold parameter are found to deteriorate the skin-friction coefficient. An ideal value for the threshold parameter has been found to be around 2. The unsteady calculations has been validated with a pitching NACA 0012 airfoil. The capability of the code to handle bodies in relative motion has been demonstrated by a store separation problem.

Acknowledgments

This work is partially funded by the U.S. Air Force Office of Scientific Research and Teledyne Brown Engineering Company.

References

[1] Lohner, R., "Adaptive Remeshing for Transient Problems with Moving Bodies," AIAA Paper 88-3737, 1988.

[2] Marcum, D. L., and Weatherill, N. P., "Aerospace Applications of Solution Adaptive Finite Element Analysis," *Computer Aided Geometric Design*, Vol. 12, No. 7, 1995, pp. 709–731.

[3] Mavriplis, D. J., and Jameson, A., "Multigrid Solution of the Navier–Stokes Equation on Triangular Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1415–1425.

[4] Morgon, K., Peraire, J., and Peiro, J., "The Computation of Three Dimensional Flows Using Unstructured Grids," *Computer Methods in Applied Mechanics and Engineering*, Vol. 87, No. 2–3, 1991, pp. 335–352.

[5] Venkatakrishnan, V., and Mavriplis, D. J., "Computation of Unsteady Flows over Complex Geometries in Relative Motion," *Proceedings of the First AFOSR Conference on Dynamic Motion CFD*, Rutgers Univ., New Brunswick, NJ, 1996, pp. 93–112.

[6] Whitaker, D. L., Slack, D. C., and Walters, W., "Solution Algorithms for the Two-Dimensional Euler Equations on Unstructured Meshes," AIAA Paper 90-0697, Jan. 1990.

[7] Whitfield, D. L., Janus, J, M., and Simpson, L. B., "Implicit Finite Volume High Resolution Wave Split Scheme for Solving the Unsteady Three Dimensional Euler and Navier–Stokes Equations on Stationary or Dynamic Grids," Engineering and Industrial Research Station, Mississippi State Univ., Rept. MSSU-EIRS-ASE-88-2, Mississippi State, MS, Feb. 1988.

[8] Dannenhoffer, J. F., "Automatic Generation of Block Structures—Progress and Challenges," *Proceedings of 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, edited by B. K. Soni, J. F. Thompson, J. Hauser, and P. Eiseman, National Science Foundation Engineering Research Center for Computational Field Simulation, Mississippi State Univ., Mississippi State, MS, 1996, pp. 403–412.

[9] Thompson, J. F., "A General Three-Dimensional Elliptic Grid Generation System on a Composite Block Structure," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, 1987, pp. 377–416.

[10] Lohner, R., and Parikh, P., "Generation of Three Dimensional Unstructured Grids by the Advancing Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 10, 1988, pp. 1135–1149.

[11] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625.

[12] Mavriplis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 92-49, Hampton, VA, Oct. 1992.

[13] Weatherill, N. P., "Grid Generation by Delaunay Triangulation," von Kármán Inst. for Fluid Dynamics Lecture Series, 1993–1994.

[14] Cooper, G. K., and Sirbough, J. R., "PARC Code: Theory and Usage," Arnold Engineering Development Center, AEDC-TR-89-15, Arnold AFB, TN, 1989.

[15] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper 95-0212, Jan. 1995.

[16] Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper 94-0417, Jan. 1994.

[17] Baum, J. D., Luo, H., and Lohner, R., "A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies," AIAA Paper 94-0414, Jan. 1994.

[18] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," AIAA Paper 94-1926, June 1994.

[19] Nakahashi, K., Nozaki, O., Kikuchi, K., and Tamura, A., "Navier–Stokes Computations of Two- and Three-Dimensional Cascade Flows," *Journal of Propulsion and Power*, Vol. 5, No. 3, 1989, pp. 320–326.

[20] Lohner, R., "Matching Semistructured and Unstructured Grids for Navier–Stokes Calculations," AIAA Paper 93-3338, July 1993.

[21] Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," AIAA Paper 96-2000, June 1996.

[22] Kallinderis, Y., Khawaja, A., and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," AIAA Paper 95-0211, Jan. 1995.

[23] Weatherill, N. P., "Mixed Structured–Unstructured Meshes for Aerodynamic Flow Simulation," *Aeronautical Journal*, Vol. 94, No. 934, 1990, pp. 111–123.

[24] Kao, K.-H., and Liou, M.-S., "Advance in Overset Grid Schemes: From Chimera to DRAGON Grids," *AIAA Journal*, Vol. 33, No. 10, 1995, pp. 1809–1815.

[25] Huang, C.-T., "Hybrid Grid Generation System," M.S. Thesis, Dept. of Aerospace Engineering, Mississippi State Univ., Mississippi State, MS, Aug. 1996.

[26] Mathur, R., Madavan, N. K., and Rajagopalan, R. G., "Solution Adaptive Structured–Unstructured Grid Method for Unsteady Turbomachinery Analysis, Part 1: Methodology and Part 2: Results," *Journal of Propulsion and Power*, Vol. 10, No. 4, 1994, pp. 576–592.

[27] Tsung, F. L., Loellbach, J., Kwon, O., and Hah, C., "A Three-Dimensional Structured/Unstructured Hybrid Navier–Stokes Method for Turbine Blade Rows," AIAA Paper 94-3369, June 1994.

[28] Parthasarathy, V., Kallinderis, Y., and Nakajima, K., "Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic-Tetrahedral Meshes," AIAA Paper 95-0670, Jan. 1995.

[29] Roe, P. L., "Approximate Riemann Solvers, Parameter Vector, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[30] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Non-Symmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[31] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1992.

[32] Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[33] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, Jan. 1993.

[34] Whitfield, D. L., and Taylor, L., "Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes," AIAA Paper 91-1539, June 1991.

[35] Whitfield, D. L., "An Implicit Solution of Roe's Approximate Riemann Solver Using True and Approximate Jacobians, Including Discretized and Sparse Matrix Update," Engineering and Industrial Research Station, Mississippi State Univ., MSSU-EIRS-ERC-93-1, Mississippi State, MS, Oct. 1992.

[36] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Series in Computational Methods in Mechanics and Thermal Sciences, Hemisphere, New York, 1984, p. 189.

[37] Warsi, Z. U. A., *Fluid Dynamics: Theoretical and Computational Approaches*, CRC Press, Boca Raton, FL, 1993, pp. 529–531.

[38] Koomullil, R. P., "Flow Simulation System for Generalized Static and Dynamic Grids," Ph.D. Dissertation, Dept. of Aerospace Engineering, Mississippi State Univ., Mississippi State, MS, May 1997.

[39] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and Its Applications to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[40] Singh, K. P., Newman, J. C., and Baysal, O., "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," *AIAA Journal*, Vol. 33, No. 4, 1995, pp. 641–649.

[41] Landon, R. H., "NACA0012 Oscillating and Transient Pitching," Data Set 3, R-702, Compendium of Unsteady Aerodynamic Measurements, AGARD, Aug. 1982.

J. Kallinderis
Associate Editor