AIAA JOURNAL
Vol. 30, No. 5, May 1992

Spatial Adaptation of Unstructured Meshes for Unsteady Aerodynamic Flow Computations

Russ D. Rausch*
Purdue University, West Lafayette, Indiana 47907
John T. Batina†
NASA Langley Research Center, Hampton, Virginia 23665
and
Henry T. Y. Yang‡
Purdue University, West Lafayette, Indiana 47907

Spatial adaptation procedures for the accurate and efficient solution of steady and unsteady inviscid flow problems are described. The adaptation procedures were developed and implemented within a two-dimensional unstructured-grid upwind-type Euler code. These procedures involve mesh enrichment and mesh coarsening to either add points in high gradient regions of the flow or remove points where they are not needed, respectively, to produce solutions of high spatial accuracy at minimal computational cost. The paper describes a new approach to detect high gradient regions of the flow that uses the substantial derivative of density. Additionally, the paper gives a detailed description of the enrichment and coarsening procedures and further describes a new weighted averaging procedure and the interpolation of flow variables that improves the overall accuracy of the flow solver. Presented are comparisons with alternative results and experimental data to provide an assessment of the accuracy and efficiency of the capability. Steady and unsteady transonic results, obtained using spatial adaptation for the NACA 0012 airfoil, are shown to be of high spatial accuracy, primarily in that the shock waves are very sharply captured. The results were obtained with large computational savings when compared to results for a globally enriched mesh with cells subdivided as many times as the finest cells of the adapted grid.

Introduction

CONSIDERABLE progress has been made over the past decade in developing methods of dynamically adapting computational meshes based on the numerical solution of partial differential equations.[1] These methods are being developed to produce higher spatial accuracy in such solutions more efficiently. Spatial accuracy is obviously important when modeling continuous equations with a discrete set of points. It is generally understood, of course, that accuracy is improved when the number of mesh points in a fixed computational domain is increased. Associated with an increase in the number of mesh points, however, are increased computer run times and memory costs. Hence, for efficiency, it is important to enrich meshes *locally* based on the numerical solution, in contrast to using *globally* fine meshes, to minimize the total number of mesh points and hence minimize the cost for a given spatial accuracy. The methods of mesh adaptation can be separated into three general categories: 1) mesh regeneration, 2) mesh movement, and 3) mesh enrichment. The first method, mesh regeneration, places the work of adapting the mesh on the mesh generation program rather than on the actual numerical solution procedure of the governing equations. In this method, a solution is first obtained, and regions of relatively large discretization errors are detected. A new mesh is then generated to concentrate points in regions where the large discretization errors occur. This new mesh may contain more or fewer points than the original mesh. For the second method, mesh movement, the number of points in the computational domain remains fixed. To resolve more accurately the solution spatially, these points are moved into regions where solution gradients are relatively large. In general, this can be accomplished by two different ways. The first way models the mesh as a spring network, where points are joined by linear springs with spring stiffnesses proportional to solution gradients. The mesh is then allowed to move into the relatively high gradient regions to produce effectively a locally finer mesh. The second way uses forcing functions in a Poisson-equation grid generator to redistribute points. Either method of mesh movement is easily implemented within existing solution algorithms because only the locations of the existing mesh points are changed. The final method of spatial adaptation is mesh enrichment. In this method, points are added to regions of relatively large solution error by dividing locally the cells that make up the mesh or by embedding finer meshes in these regions. This method differs from the mesh regeneration and movement approaches in which the mesh is made finer in local regions while the global mesh topology remains the same. The method of mesh enrichment is also generally regarded as having advantages over regeneration and movement, especially for transient problems, because of the higher degree of flexibility in being able to add points where they are needed and to remove points where they are not needed. This higher degree of flexibility, however, requires the need to redetermine the "structure" of the mesh after each mesh adaptation. Therefore, a disadvantage of mesh

Received Feb. 22, 1991; presented as Paper 91-1106 at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference, Baltimore, MD, April 8–10, 1991; revision received Aug. 20, 1991; accepted for publication Aug. 21, 1991.

*Graduate Research Assistant, School of Aeronautics and Astronautics. Member AIAA.
†Senior Research Scientist, MS 173, Unsteady Aerodynamics Branch, Structural Dynamics Division. Senior Member AIAA.
‡Professor and Dean, Schools of Engineering. Fellow AIAA.

enrichment is that implementation of the method involves significant modifications to existing solution algorithms.

For the Euler and Navier-Stokes equations, computational fluid dynamics algorithms are being developed based on spatial adaptation methods. With these equations, relatively large spatial discretization errors may be encountered with flow features such as shock waves, shear layers, boundary layers, and expansion fans. These flow features can be resolved more accurately using the adaptation methods mentioned above. Nakahashi and Deiwert,[2] for example, have used tension and torsional springs to move the mesh into regions where relatively large spatial discretization errors occur. This mesh movement approach showed considerable versatility for the problems treated. However, various constants were needed to control orthogonality and smoothness, and direct control of an optimal mesh adaptation procedure was generally not possible. Further examples of spatial adaptation methods include the work of Usab and Murman.[3] In Ref. 3, embedded meshes of quadrilateral cells and nodes were used in regions of the mesh where shock waves occurred. This approach improved the spatial accuracy of the numerical method, which resulted in highly accurate solutions for steady flow problems. Dannenhoffer and Baron[4,5] extended the work in this area using irregularly shaped embedded regions, which were coupled to the base mesh by a multiple-grid solution algorithm. Several other examples of spatial adaptation include methods that use flow solvers based on unstructured triangular and tetrahedral meshes in two and three dimensions, respectively. Peraire et al.[6,7] used mesh regeneration coupled with a finite element solution algorithm to sharply capture shock waves and complex shock structures. Löhner and co-workers[8-13] have developed a procedure to enrich the mesh for transient flow problems locally by dividing elements that make up a base mesh to capture shock waves. Furthermore, in this procedure, elements may be removed (coarsened) from the mesh if they are not necessary to produce a given level of spatial accuracy.

With respect to solution algorithms based on unstructured meshes, the results published by the authors[14] demonstrated that these algorithms produce steady and unsteady solutions of comparable accuracy to results obtained using structured-grid solution algorithms. The two sets of results presented in Ref. 14 were obtained using meshes of comparable density, and mesh adaptation was not used. Solutions of higher spatial accuracy are, indeed, possible through the use of mesh adaptation. Therefore, the purpose of this paper is to report on modifications to the two-dimensional unstructured-grid upwind-type Euler code of Batina[15] to include mesh enrichment and coarsening procedures. The objectives of the research are as follows: 1) to develop time-accurate enrichment and coarsening procedures for spatial adaptation; 2) to test the procedures by performing steady and unsteady calculations for a variety of cases; 3) to determine the accuracy of the spatially adapted solutions by making comparisons with published solutions produced by alternative methods and existing experimental data; and 4) to assess the efficiency of the spatially adapted solutions by making comparisons of required computer resources. The eventual goal is to develop a highly accurate and efficient solution algorithm for the Euler and Navier-Stokes equations for aeroelastic analysis of complex aircraft configurations. The paper describes a new approach to detect high gradient regions of the flow that uses the substantial derivative of density. Additionally, the paper gives a detailed description of the enrichment and coarsening procedures, a brief description of the solution algorithm of Ref. 15, and further describes a new weighted averaged procedure for the interpolation of flow variables that improves the overall accuracy of the flow solver. Steady and unsteady transonic results are presented for the NACA 0012 airfoil to demonstrate applications of the spatial adaptation procedures. The unsteady flow results for the NACA 0012 airfoil were obtained for the airfoil pitching harmonically about the quarter chord.

Upwind-Type Euler Solution Algorithm

The unsteady Euler equations are solved using the two-dimensional upwind-type solution algorithm developed by Batina.[15] The solution algorithm, which is a cell-centered finite-volume scheme,[15-17] uses upwind differencing based on flux-vector splitting similar to upwind schemes developed for use on structured meshes. Thus, the present unstructured grid algorithm is referred to as an upwind-type solution algorithm. The spatial discretization of this algorithm involves a so-called flux-split approach based on the flux-vector splitting of van Leer.[18] The flux-split discretization accounts for the local wave-propagation characteristics of the flow and captures shock waves sharply with at most one grid point within the shock structure. A further advantage is that the discretization is naturally dissipative and consequently does not require additional artificial dissipation terms or the adjustment of free parameters to control the dissipation. However, in calculations involving higher order upwind schemes, oscillations in the solution near shock waves are expected to occur. To eliminate these oscillations, flux limiting is usually required. In the present study, a continuously differentiable flux limiter was employed.[15-17]

The Euler equations are integrated in time using an implicit time-integration scheme involving a Gauss-Seidel relaxation procedure.[15] The relaxation procedure is implemented by reordering the elements that make up the unstructured mesh from upstream to downstream. The solution is obtained by sweeping twice through the mesh as dictated by stability considerations. The first sweep is performed in the direction from upstream to downstream and the second sweep is from downstream to upstream. For purely supersonic flows, the second sweep is unnecessary. This relaxation scheme is stable for large time steps and allows the selection of the step size based on the temporal accuracy required for the problem being considered, rather than on the numerical stability of the algorithm. Consequently, very large time steps may be used for rapid convergence to steady state and an appropriate step size may be selected based on temporal convergence for unsteady cases, independent of numerical stability issues.

Weighted Averaging

A weighted averaging procedure is used in the two-dimensional upwind-type Euler solution algorithm to interpolate the cell-centered values of the primitive variables to the nodes. The interpolation of cell-centered values to the mesh nodes is required for the upwind-biased interpolation of primitive variables in the Euler solution algorithm.[15] As illustrated in Fig. 1, the weighted average of the cell-centered values q_i surrounding a node is given by

$$q_0 = \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

where the nodal value q_0 is computed using the cell weights w_i. Several different weights have been used in this procedure

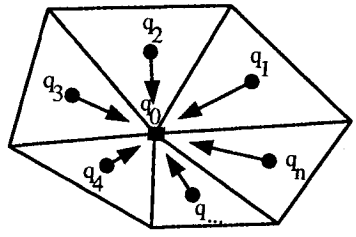


Fig. 1 Difference star for computing the weighted average q_0 of surrounding cell-centered values $q_{1,\ldots,n}$.

including the areas of surrounding cells and the reciprocal distance from the node to the locations of cell centers. Because of the disparity in cell sizes that occur when using spatial adaptation, the most accurate weighted averaging is desired. Therefore, a different approach for computing the weights has been developed based on the work of Holmes and Connell.[19] In Ref. 19, artificial dissipation for a node-based Navier-Stokes solution algorithm was computed using a Laplacian operator. The Laplacian $L(\phi_0)$ at a node was computed by

$$L(\phi_0) = \sum_{i=1}^{n} w_i(\phi_i - \phi_0) \tag{2}$$

using the surrounding nodal values ϕ_i and the weights w_i. The weights were derived by first noting that the Laplacian of a linear function is exactly zero. Because this is a desirable property to require of the computed Laplacian, the weights w_i in Eq. (2) were derived with this objective. Therefore, the spatial locations (x_i,y_i) of ϕ_i were substituted into Eq. (2) resulting in

$$L(x_0) = \sum_{i=1}^{n} w_i(x_i - x_0) = 0 \tag{3}$$

$$L(y_0) = \sum_{i=1}^{n} w_i(y_i - y_0) = 0 \tag{4}$$

The weights were then determined by defining

$$w_i = 1 + \Delta w_i \tag{5}$$

where

$$C = \sum_{i=1}^{n} (\Delta w_i)^2 \tag{6}$$

is a cost function. The cost function C was minimized to keep the weights close to unity by solving an optimization problem given the constraints of Eqs. (3) and (4). The optimization problem was solved using the method of Lagrange multipliers where Δw_i was given by

$$\Delta w_i = \lambda_x(x_i - x_0) + \lambda_y(y_i - y_0) \tag{7}$$

The solution yielded Lagrange multipliers, defined by

$$\lambda_x = \frac{(I_{xy}R_y - I_{yy}R_x)}{(I_{xx}I_{yy} - I_{xy}^2)} \tag{8}$$

$$\lambda_y = \frac{(I_{xy}R_x - I_{xx}R_y)}{(I_{xx}I_{yy} - I_{xy}^2)} \tag{9}$$

where

$$R_x = \sum_{i=1}^{n} (x_i - x_0) \tag{10}$$

$$R_y = \sum_{i=1}^{n} (y_i - y_0) \tag{11}$$

$$I_{xx} = \sum_{i=1}^{n} (x_i - x_0)^2 \tag{12}$$

$$I_{yy} = \sum_{i=1}^{n} (y_i - y_0)^2 \tag{13}$$

$$I_{xy} = \sum_{i=1}^{n} (x_i - x_0)(y_i - y_0) \tag{14}$$

Note that the weights were computed solely from the distance between node locations and no special treatment was required near mesh boundaries.

Modifications were made to this procedure for computing the Laplacian in order to develop a new procedure for computing the weights in Eq. (1). The first modification was to express the Laplacian at the nodes using surrounding cell-centered values, as illustrated in Fig. 1, rather than the surrounding nodal values. Now the Laplacian at the node can be expressed by

$$L(q_0) = \sum_{i=1}^{n} w_i(q_i - q_0) \tag{15}$$

where q_0 represents the unknown value at the node and q_i are the known values at the surrounding cell-centers. The weights w_i in Eq. (15) are now computed using the node location (x_0,y_0) and the surrounding cell-center locations (x_i,y_i). Recalling that the computed Laplacian for a linear function is zero, then for a linear variation of q

$$\sum_{i=1}^{n} w_i(q_i - q_0) = 0 \tag{16}$$

which can be rewritten in the form given by Eq. (1). Therefore, using the weights from the Laplacian procedure, Eq. (1) is an exact interpolation of the nodal value q_0 for a linear variation of cell-centered values q_i. For a nonlinear variation of cell-centered values q_i, the weighted averaging procedure using the weights from the Laplacian procedure is second-order accurate in space. Although the procedure is computationally more expensive than area or reciprocal distance weighted averaging, the increased cost is relatively small compared to the cost of solving the Euler equations.

Spatial Adaptation Procedures

In this section, the spatial adaptation procedures are described in detail. These descriptions include explanations of the flow feature detection and the procedures used to enrich and coarsen the mesh.

Flow Feature Detection

The first step of the spatial adaptation procedure is the detection of regions of relatively large discretization error so that the computational mesh can be locally enriched to improve the spatial accuracy or locally coarsened to reduce the computational costs. These regions generally occur near flow features such as shock waves, stagnation points, slip lines, and expansion fans for the Euler equations, where the dominant flow feature for the cases considered in this study is a shock wave.

There are a number of flow parameters that can be used for enrichment indicators based on the detection of shock waves. Parameters such as density, pressure, or total velocity are useful because these quantities are discontinuous through shocks. For example, first or second, divided or undivided differences in one of these parameters, similar to the work by Dannenhoffer and Baron,[4] can be used to detect shock waves. A specific example of an enrichment indicator is the magnitude of the density gradient $|\nabla\rho|$, which is often used to detect shock waves for steady flow problems. However, for unsteady flows, a measure of the temporal as well as the spatial variation of the solution is needed. For this reason, the absolute value of the substantial derivative of density $|D\rho/Dt|$ was used as an enrichment indicator, because it is the sum of the local rate of change $\partial\rho/\partial t$ as well as the convective rate

of change $\boldsymbol{u} \cdot \nabla\rho$ of density. The substantial derivative of density can also be written as

$$\frac{D\rho}{DT} = \frac{\partial\rho}{\partial t} + \nabla \cdot \rho\boldsymbol{u} - \rho\nabla \cdot \boldsymbol{u} \tag{17}$$

where the first two terms on the right-hand side of Eq. (17) are simply the continuity equation. The third term can be viewed as a simplified continuity equation that assumes steady, incompressible, linear flow. Therefore, the difference between these two continuity equations results in a measure of the: 1) unsteadiness, 2) compressibility, and 3) nonlinearities of the flow. Because the first two terms on the right-hand side of Eq. (17) are zero from the continuity equation, the substantial derivative of density can be rewritten as

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \boldsymbol{u} \tag{18}$$

For unsteady flows, the substantial derivative works well for detecting developing shock waves; whereas, the more commonly used enrichment indicators based on the instantaneous solution, such as first or second differences in density, tend to miss the initial shock wave formation. This is especially true for cases where the shock waves periodically appear and disappear in time. Other indicators, based on the substantial derivative of x-momentum ρu, y-momentum ρv, and energy e have also been investigated. These substantial derivatives are written as

$$\frac{D\rho u}{Dt} = \left\{\frac{\partial\rho u}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\rho uv}{\partial y}\right\} - \left\{\frac{\partial p}{\partial x} + \rho u\nabla \cdot \boldsymbol{u}\right\} \tag{19}$$

$$\frac{D\rho v}{Dt} = \left\{\frac{\partial\rho v}{\partial t} + \frac{\partial\rho uv}{\partial x} + \frac{\partial(\rho v^2 + p)}{\partial y}\right\} - \left\{\frac{\partial p}{\partial y} + \rho v\nabla \cdot \boldsymbol{u}\right\} \tag{20}$$

$$\frac{De}{Dt} = \left\{\frac{\partial e}{\partial t} + \frac{\partial(e + p)u}{\partial x} + \frac{\partial(e + p)v}{\partial y}\right\} - \{\nabla \cdot (\rho\boldsymbol{u}) + e\nabla \cdot \boldsymbol{u}\} \tag{21}$$

where once again each derivative results in the difference between a complete conservation equation and an approximate equation. The first expression in brackets on the right-hand side of Eqs. (19–21) is zero for the Euler x-momentum, y-momentum, and energy equations, respectively. The last expression in brackets is an approximate equation of the respective conservation laws. For example, the second equation in brackets in Eq. (19) can be derived from the conservation of x-momentum equation if the equation is simplified by assuming the local rate of change $\partial/\partial t$ and the convective rate of change $\boldsymbol{u} \cdot \nabla$ of x-momentum is zero. Therefore, as one might expect, if the absolute value of Eq. (19) is used as an enrichment indicator it would detect regions of the flow where the x-momentum is most rapidly changing. Similarly, Eqs. (20) and (21) would detect the rate of change of y-momentum and energy, respectively. To evaluate each expression as an enrichment indicator, the substantial derivatives of the conserved variables were computed for a transient shock problem. In this study, the performance of each of the substantial derivatives in detecting shock waves was found to be similar. It is not surprising, however, because all of the indicators involve the same term $\nabla \cdot \boldsymbol{u}$. Therefore, the enrichment indicator used in this study was the absolute value of the substantial derivative of density [Eq. (18)].

Mesh Enrichment

Generally, mesh enrichment is performed by starting with a relatively coarse mesh of cells and then subdividing these cells until a given level of spatial accuracy has been obtained. To prevent cells from being enriched too many times near flow discontinuities such as shock waves, an upper bound is placed on the number of times a cell can be divided. For transient problems the mesh enrichment procedure may be performed at each time step of the integration of the governing flow equations or it may be performed once every set number of time steps.

Mesh enrichment is performed by using the enrichment indicator to determine if a cell is to be subdivided into smaller cells. To accomplish this, the enrichment indicator is computed for each cell and compared with a preset threshold value to determine whether a cell should be subdivided. If the threshold is exceeded, a new node is created at the midpoint of each edge of the triangular cell, and the cell is subdivided into four smaller cells. Special care must be taken, however, when an edge that is to be bisected lies on a boundary of the mesh, because the midpoint of the edge does not generally lie on the boundary. In this case, the location of the new node is determined by using a parametric spline of the boundary coordinates. Furthermore, the values of the flow variables for the new cells are determined from a linear interpolation of conserved variables located at the nodes and the cell centers of the original cells.

For a given cell to be enriched, either one edge or all three edges are bisected. In the event that only two edges are marked to be bisected, the third edge is automatically bisected to prevent the creation of highly skewed or stretched cells. Each time the mesh is enriched, a cell may be divided in one of two ways, as shown in Fig. 2, a and b. The first way, shown in Fig. 2a, results when all three edges of a cell have been marked for enrichment. In this situation the cell is divided into four new cells where the vertices of the inner cell are in general midpoints of edges that make up the original cell. The original cell is, thus, referred to as a type-4 element because after enrichment it becomes four new triangular cells. The second way, shown in Fig. 2b, occurs if only one edge of the base cell is marked for enrichment. In this situation, the marked edge is bisected, and two new cells are formed. The original cell is, thus, referred to as a type-2 element because after enrichment it becomes two new cells. New cells formed from a type-4 element may be enriched further. However, to prevent highly stretched cells, cells formed from a type-2 element are restricted from being divided further as indicated in Fig. 2, c and d. For cells from a type-2 element,

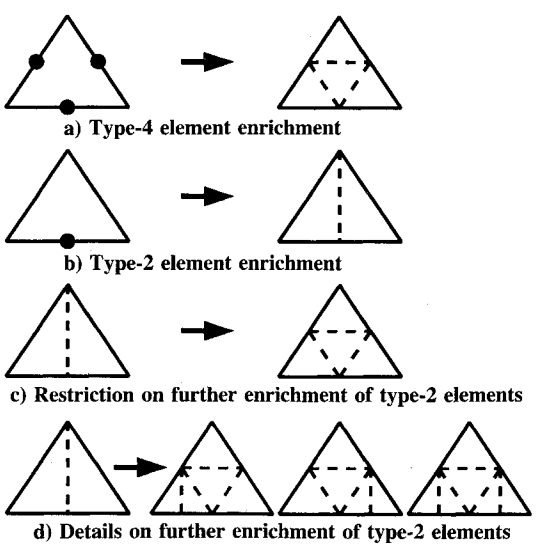


Fig. 2 Diagrams illustrating mesh enrichment possibilities.

if any of the five edges that make up the two new cells are marked for enrichment, the original cell is made into a type-4 element as shown in Fig. 2c. If in addition to this, either or both of the bottom two edges are marked (the lower left, right, or both), cells of the type-4 element are further enriched accordingly, as shown in Fig. 2d.

Mesh Coarsening

Generally, mesh coarsening removes added nodes and cells from previously enriched meshes to delete them from local regions of the mesh where certain flow features are no longer present. This procedure is necessary to adapt meshes efficiently to the numerical solution of the governing flow equations in order to minimize computational cost. The mesh coarsening procedure starts by marking all of the cells that do not have edges that are marked to be bisected from the mesh enrichment procedure. Figure 3 shows an example of the coarsening procedure where the dashed lines represent cells formed by a previous enrichment of the mesh. In Fig. 3a, cells that are candidates for removal are denoted by a "1." The marked cells are then used to determine nodes that are candidates for removal. As shown in Fig. 3b, the candidate nodes that may be removed are nodes that are surrounded by cells that are candidates for removal (identified by the "1s" in Fig. 3a). Cells that are to be removed from the list of candidates are determined by searching the tree data structure that stores the mesh enrichment history. Cells for removal are cells that came from type-2 or type-4 elements and are subsequently marked for removal. Once the nodes and elements have been selected they are subjected to a final simultaneous evaluation for actual removal from the mesh. Each time the mesh is coarsened, cells and nodes may be removed in one of several ways as shown in Fig. 4. (In the figure, nodes to be removed are indicated by the open circles and fixed nodes are indicated by the closed circles.) For a type-4 element, if all three of the nodes that form the inner triangle are candidate nodes, then the nodes are removed eliminating the inner triangle as shown in Fig. 4a. This leaves only the one original cell that was previously divided into four. Similarly, if two of the three nodes that form the inner triangle are candidate nodes, the two nodes are removed and a type-2 element is formed as shown in Fig. 4b. If only one of the candidate nodes is marked then nothing is done. For a type-2 element there is only one node that may be removed, which is the midpoint of a previously bisected edge. Removal of this node leaves only the cell that was originally divided into two as shown in Fig. 4c.

Temporal Convergence Considerations

Careful consideration must be given to the time step chosen to ensure temporal convergence of unsteady problems, especially when spatial adaptation procedures are used. In general the temporal convergence of the numerical solution of a

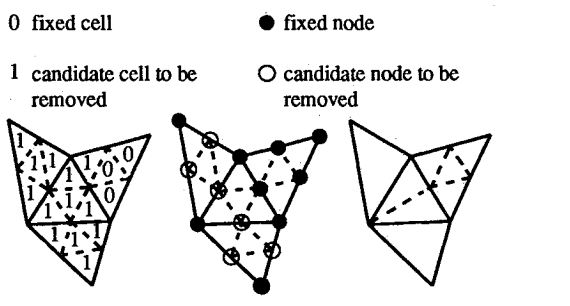


Fig. 3 Diagrams illustrating details of the mesh coarsening procedure.

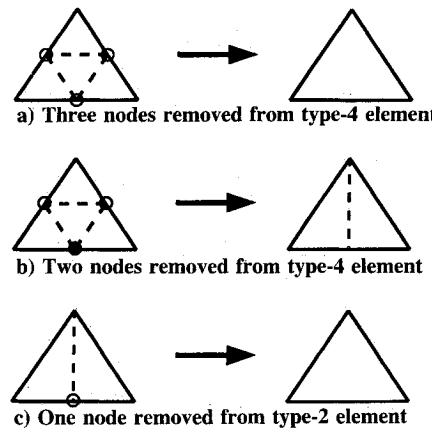


Fig. 4 Diagrams illustrating mesh coarsening possibilities.

time accurate problem is difficult to quantify and is related to the nondimensional time step Δt, a characteristic length (A/s), and the local maximum characteristic wave speed ($|u| + a$) of each cell in the computational mesh. This relation is given by the Courant-Friedrichs-Lewy (CFL) number for each cell

$$CFL \approx \Delta t \frac{s}{A} (|u| + a) \tag{22}$$

where s is the length of a cell edge, A is the area of the cell, u is a local flow speed normal to a cell edge, and a is the local speed of sound. For unsteady problems, experience has shown that the local CFL number must remain below approximately 20 for solutions to be converged in time. However, the local allowable CFL number is highly dependent on the time-marching procedure used. For example, either explicit or implicit time-marching procedures may be used to integrate the governing flow equations in time for unsteady problems. Because the allowable CFL number based on a stability analysis of an explicit time-marching procedure is usually small compared to 20, the time step is restricted to be relatively small. Therefore, the numerical stability of an explicit time-marching procedure dictates the time step that can be used. For an implicit time-marching procedure, however, numerical stability of the procedure is, in general, not an issue. Hence, the time step should be based on minimizing the effects of linearization errors, minimizing the effects of relaxation errors (or factorization errors in the case of a factored scheme), and obtaining a temporally converged solution. Irrespective of the time-marching procedure used, the spatial adaptation procedures have a significant effect on the allowable time step when the CFL number is restricted by either the temporal accuracy of the solution or the numerical stability of the solution algorithm. This effect can be demonstrated in a simple example of mesh enrichment of a single cell. Suppose, for example, the local CFL number for a cell is restricted to 20 for a temporally converged solution and the cell is enriched one level, where it is divided into four smaller cells each with area $A/4$ as shown in Fig. 2a. Likewise, each edge of the cell is bisected resulting in each of the new cells having edge lengths of $s/2$. Because, from Eq. (22), the time step is proportional to the characteristic length (A/s), the allowable time step is halved when the cell is enriched one level:

$$\Delta t \propto \frac{A/4}{s/2} = \frac{1}{2}\frac{A}{s} \tag{23}$$

For unsteady problems involving a pitching airfoil in forced harmonic motion, the effect of enrichment on the number of time steps per cycle of motion can also be demonstrated. First

the number of time steps per cycle of motion N is related to the time step by

$$N = \frac{\pi}{kM_\infty \Delta t} \tag{24}$$

where k is the reduced frequency and M_∞ is the freestream Mach number (because Δt is nondimensionalized by the freestream speed of sound). Therefore, the number of time steps per cycle of motion is inversely related to the time step:

$$N \propto \frac{1}{\Delta t} \tag{25}$$

Because the time step is halved as the mesh is enriched one level, the number of time steps required per cycle of motion doubles for a similar level of temporal convergence. Hence, the relationship between the number of time steps per cycle of motion N required on a mesh to the number of time steps per cycle of motion N_n on the same mesh using n levels of mesh enrichment can be stated as

$$N_n \approx 2^n N \tag{26}$$

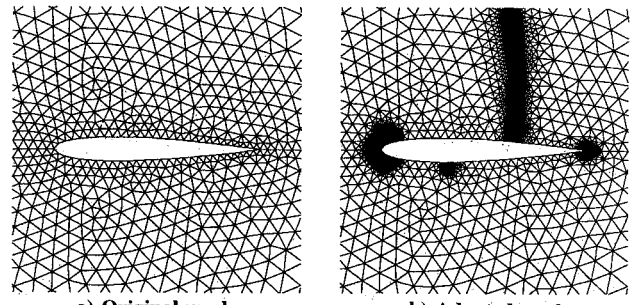
As an example, suppose an unsteady problem on a coarse mesh requires 1250 time steps per cycle of motion for a temporally converged solution. The same mesh and case using three levels of enrichment would require approximately 10,000 time steps per cycle of motion.

Results and Discussion

Calculations were performed for the NACA 0012 airfoil to assess the accuracy and efficiency of the spatial adaptation procedures. Steady and unsteady airfoil results are presented for comparison with published solutions using alternative methods and experimental data to determine the accuracy of the results. Timing comparisons are also made between results obtained using globally (estimated) and locally adapted meshes to determine the computational savings obtained by using the spatial adaptation procedures.

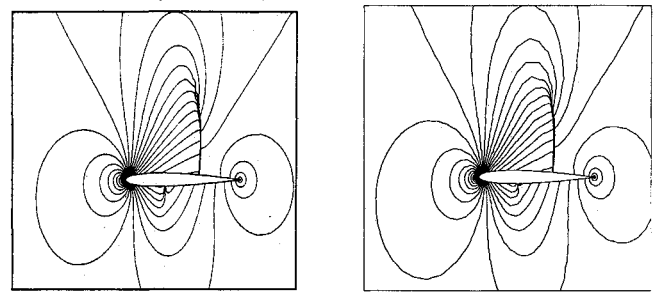
Steady Results

Steady flow results were obtained for the NACA 0012 airfoil for comparison with results reported by Pulliam and Barton.[20] The results presented are for a NACA 0012 airfoil at a freestream Mach number of $M_\infty = 0.8$ and an angle of attack of $\alpha_0 = 1.25$ deg. The starting mesh and final spatially adapted mesh are presented along with corresponding pressure contours for comparison with results of Ref. 20. The starting coarse mesh for the calculation, shown in Fig. 5a, contains 1854 nodes and 3628 cells and extends 20 chord lengths to a circular outer boundary. The solution was obtained by adapting the mesh to the solution every 500 iterations for the first 2000 iterations (four levels of enrichment), and then the solution was converged to machine zero on the final adapted mesh. The final mesh after four levels of enrichment, shown in Fig. 5b, contains 8947 nodes and 17,653 cells. For this case, there is a relatively strong shock wave on the upper surface of the airfoil near 62% chord and a relatively weak shock wave on the lower surface near 30% chord. Results from Ref. 20 were computed on a structured mesh of C-type topology containing 560 × 65 mesh points where grid lines were clustered above and below the airfoil in the shock regions to produce a more accurate result. A comparison of the pressure contour lines from Pulliam and Barton,[20] shown in Fig. 6a, with the contour lines ($\Delta p = 0.025$) shown in Fig. 6b, obtained with the present spatial adaptation procedure, indicates excellent agreement, which verifies the accuracy of the adaptation procedures. Finally, the computed lift, drag, and moment coefficients from the present calculation are $c_l = 0.3600$, $c_d = 0.0238$, and $c_m = -0.0390$, respectively. A comparison



a) Original mesh b) Adapted mesh

Fig. 5 Comparison of meshes used in the calculation for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha_0 = 1.25$ deg.



a) Ref. 20 b) Adapted mesh

Fig. 6 Comparison of pressure contour lines from Pulliam and Barton[20] and contour lines ($\Delta p = 0.025$) obtained with the present spatial adaptation procedures for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha_0 = 1.25$ deg.

of these coefficients with those obtained by Pulliam and Barton,[20] indicates agreement to within 1% for the lift and drag, and to within 5% for the moment coefficient.

The computational savings for the steady flow problem can be estimated by comparing the number of cells in the adapted mesh to the number of cells in a globally fine mesh. The total number of cells for the globally fine mesh is computed by multiplying the number of cells in the starting coarse mesh by four (because each cell is divided into four smaller cells) raised to a power represented by the number of enrichment levels. Therefore, a globally fine mesh would contain 928,768 cells (3628×4^4), which in comparison with the final adapted mesh containing 17,653 cells, results in a computational savings of approximately 53. This savings factor does not include the computational overhead of the four mesh adaptations, which accounts for less than 1% of the total CPU time. If timing comparisons are made with solutions obtained using structured grid algorithms, the computational overhead of the indirect addressing of the unstructured grid algorithm must be accounted for. For example, the computational work for steady-state solutions obtained using unstructured grid algorithms have been shown to be two to five times more expensive than those obtained using structured grid algorithms.[21] Nonetheless, when spatial adaptation procedures are used with unstructured grid algorithms, an order of magnitude computational savings can be achieved over solutions obtained using structured grid algorithms for comparable spatial accuracy.

Unsteady Results

Unsteady results are presented for the NACA 0012 airfoil using spatial adaptation for AGARD case 5, proposed by the AGARD Structures and Materials Panel.[22] This case involves the NACA 0012 airfoil pitching harmonically about the quarter chord at a reduced frequency based on semichord of $k = 0.0814$. The calculation was performed for three cycles of motion to obtain a periodic solution using 10,000 time steps

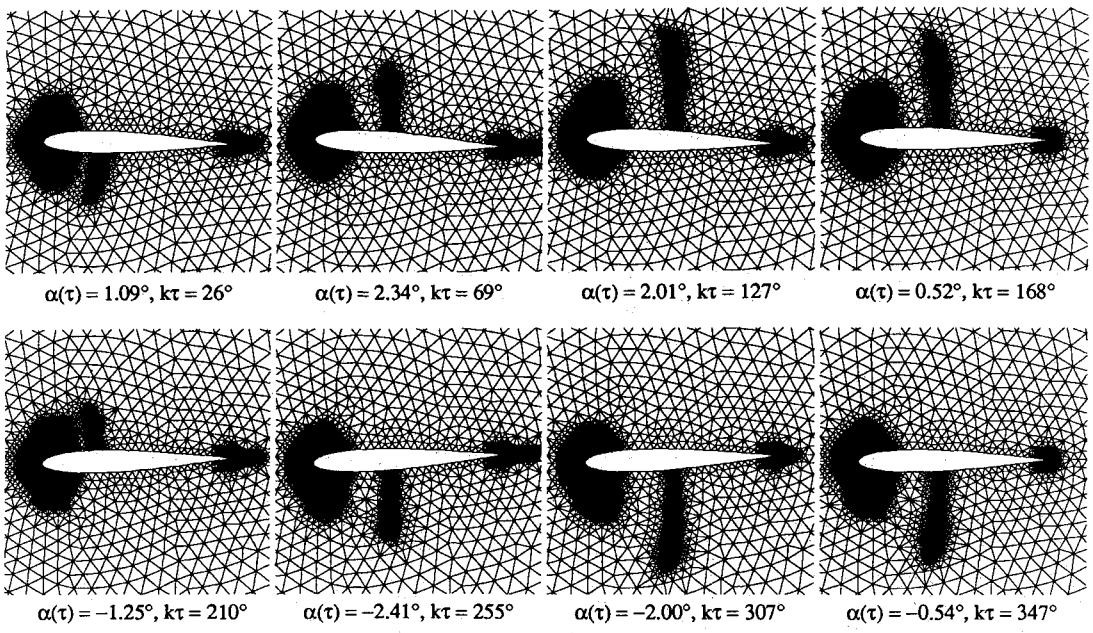


Fig. 7 Instantaneous meshes produced by the spatial adaptation procedure for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg, $\alpha_1 = 2.51$ deg, and $k = 0.0814$.

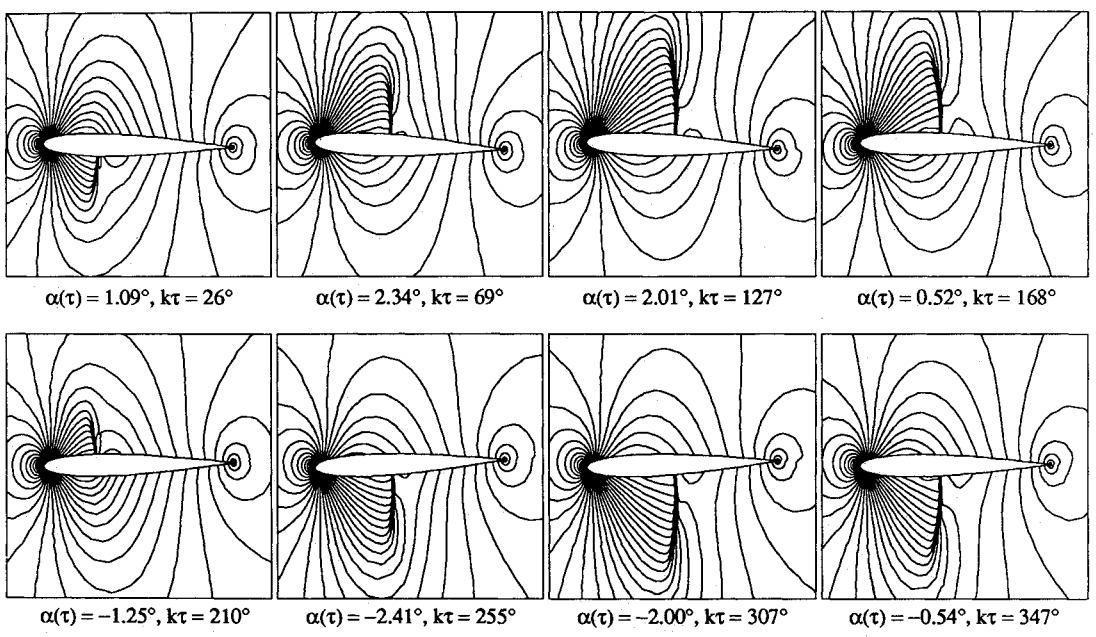


Fig. 8 Instantaneous density contour lines ($\Delta\rho = 0.02$) from the spatially adapted solution for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg, $\alpha_1 = 2.51$ deg, and $k = 0.0814$.

per cycle starting with the same coarse mesh (Fig. 5a) that was used for the nonadapted steady flow result. During the course of the calculations, the mesh was spatially adapted every 10 time steps using three levels of enrichment. Results were obtained for the airfoil pitching with an amplitude of $\alpha_1 = 2.51$ deg at $M_\infty = 0.755$ and $\alpha_0 = 0.016$ deg. Figure 7 shows the instantaneous adapted meshes and Fig. 8 shows the corresponding instantaneous density contour lines ($\Delta\rho = 0.02$). The instantaneous meshes and density contour lines during the third cycle of motion were plotted at eight points in time. In each plot, the instantaneous pitch angle $\alpha(\tau)$ and the instantaneous angular position $k\tau$ in the cycle are noted. The instantaneous meshes (Fig. 7) clearly indicate the enrichment in regions near the shock waves and near the stagnation points. They also show coarsened regions where previously enriched regions have relatively small flow gradients. The density contours during the cycle (Fig. 8) demonstrate the ability of the spatial adaptation procedures to produce sharp transient shock waves. The corresponding surface pressure distributions during the third cycle of motion are shown in Fig. 9 for comparison with experimental data.[23] In each pressure plot the instantaneous pitch angle $\alpha(\tau)$ and the angular position $k\tau$ in the cycle are noted. During the first part of the cycle there is a shock wave on the upper surface of the airfoil, and the flow over the lower surface is predominately subcritical. During the latter part of the cycle the flow about the upper surface is subcritical while a shock forms along the lower surface. The pressure distributions indicate that the shock position oscillates over approximately 25% of the chord along the upper and lower surfaces, requiring the spatial adaptation procedure to track the movement and development of the transient shocks accurately. In general, the calculated results using the spatial

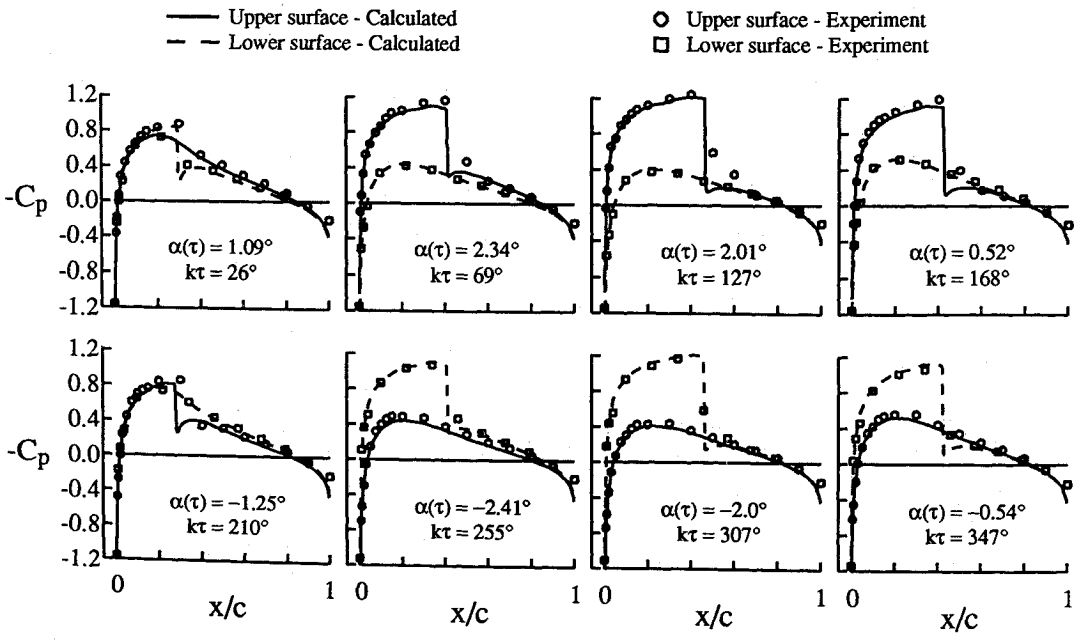


Fig. 9 Comparison of instantaneous pressure distributions for the NACA 0012 airfoil pitching harmonically at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg, $\alpha_1 = 2.51$ deg, and $k = 0.0814$ computed using spatial adaptation.

adaptation procedures compare well with the experimental data.

The computational savings obtained by using the spatial adaptation procedures can be estimated by comparing the total number of cells marched in time with the estimated number of cells marched in time if a globally fine mesh were used. The total number of cells marched in time for the globally fine mesh is computed by multiplying the number of cells in the starting coarse mesh by the number of cycles, the number of time steps per cycle, and four raised to the number of enrichment levels (because each cell is divided into four smaller cells). Therefore, for AGARD case 5, a total of 413 million cells were marched compared to 6966 million cells for the globally fine mesh resulting in a computational savings of a factor of 17. This factor does not include the computational overhead of the spatial adaptation procedure, which was approximately 7.5% of the total CPU time.

Concluding Remarks

Spatial adaptation procedures for the accurate and efficient solution of steady and unsteady inviscid flow problems were described. The adaptation procedures were developed and implemented within a two-dimensional unstructured-grid upwind-type Euler code. These procedures involve mesh enrichment and mesh coarsening to either add points in high gradient regions of the flow or remove points where they are not needed, respectively, to produce solutions of high spatial accuracy at minimal computational cost. A novel approach for detecting flow features based on the substantial derivative of density was used as an enrichment indicator. This enrichment indicator worked well for detecting developing shock waves in unsteady flows. This is a significant improvement over the more commonly used enrichment indicators based on the instantaneous solution (such as first or second differences in density) that miss the initial shock wave formation, especially for cases where the shock waves periodically appear and disappear in time.

Steady and unsteady transonic results were presented for the NACA 0012 airfoil to demonstrate applications of the spatial adaptation procedures to two-dimensional problems. The unsteady flow results were obtained for an airfoil pitching harmonically about the quarter chord. Both the steady and unsteady solutions obtained using spatial adaptation were shown to be of high spatial accuracy, primarily in that the shock waves were very sharply captured. Comparing the cost of results obtained on a spatially adapted mesh with the estimated cost of results obtained on a globally fine mesh of comparable mesh density, a computational savings of a factor of approximately 53 was achieved for the steady calculations using four levels of enrichment. Similar calculations using structured grid algorithms were estimated to be an order of magnitude more expensive for comparable spatial accuracy. Comparisons of coefficients from the spatial adaptation solution with those of Pulliam and Barton,[20] indicated agreement to within 1% of the lift and drag coefficients and to within 5% for the moment coefficient. Similarly, for the unsteady calculation, a computational savings of a factor of 17 was achieved and comparisons that were made with experimental surface pressure data indicated good agreement.

Acknowledgments

The work constitutes a part of the first author's Ph.D. thesis at Purdue University and was supported by the NASA Graduate Student Researchers program under Grant NGT-50406. Also, the authors acknowledge Ken Morgan and Jaime Peraire of the Imperial College of Science, Technology, and Medicine, London, England for providing the grid generation program that was used to generate the grid for the NACA 0012 airfoil used in the present study.

References

[1]Thompson, J. F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, AIAA Paper 84-1606, Snowmass, CO, June 1984.

[2]Nakahashi, K., and Deiwert, G. S., "Self-Adaptive-Grid Method with Application to Airfoil Flow," *AIAA Journal*, Vol. 25, No. 4, 1987, pp. 513–520.

[3]Usab, W. J., and Murman, E. M., "Embedded Mesh Solution of the Euler Equation Using a Multiple-Grid Method," AIAA Sixth Computational Fluid Dynamics Conference, AIAA Paper 83-1946-CP, Danvers, MA, July 1983.

[4]Dannenhoffer, J. F., and Baron, J. R., "Grid Adaption for the 2-D Euler Equations," AIAA 23rd Aerospace Sciences Meeting, AIAA Paper 85-0484, Reno, NV, Jan. 1985.

[5]Dannenhoffer, J. F., and Baron, J. R., "Robust Grid Adaption for Complex Transonic Flows," AIAA 24th Aerospace Sciences

Meeting, AIAA Paper 86-0495, Reno, NV, Jan. 1986.

[6]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, pp. 449–466.

[7]Peraire, J., Morgan, K., Peiro, J., and Zienkiewicz, O. C., "An Adaptive Finite Element Method for High Speed Flows," AIAA 25th Aerospace Sciences Meeting, AIAA Paper 87-0558, Reno, NV, Jan. 1987.

[8]Löhner, R., and Morgan, K., "Improved Adaptive Refinement Strategies for Finite Element Aerodynamic Computations," AIAA 24th Aerospace Sciences Meeting, AIAA Paper 86-0499, Reno, NV, Jan. 1986.

[9]Löhner, R., Laboratory for Computational Physics and Fluid Dynamics, "The Efficient Simulation of Strongly Unsteady Flows by the Finite Element Method," AIAA 25th Aerospace Sciences Meeting, AIAA Paper 87-0555, Reno, NV, Jan. 1987.

[10]Löhner, R., "An Adaptive Finite Element Solver for Transient Problems with Moving Bodies," *Computers & Structures*, Vol. 30, No. 1/2, 1988, pp. 303–317.

[11]Baum, J. D., and Löhner, R., "Numerical Simulation of Shock-Elevated Box Interaction Using an Adaptive Finite-Element Shock Capturing Scheme," AIAA 27th Aerospace Sciences Meeting, AIAA Paper 89-0653, Reno, NV, Jan. 1989.

[12]Löhner, R., "Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems," AIAA 27th Aerospace Sciences Meeting, AIAA Paper 89-0365, Reno, NV, Jan. 1989.

[13]Löhner, R., and Baum, J. D., "Numerical Simulation of Shock Interaction with Complex Geometry Three-Dimensional Structures Using a New Adaptive H-Refinement Scheme on Unstructured Grids," AIAA 28th Aerospace Sciences Meeting, AIAA Paper 90-0700, Reno, NV, Jan. 1990.

[14]Rausch, R. D., Batina, J. T., and Yang, T. Y., "Euler Flutter Analysis of Airfoils Using Unstructured Dynamic Meshes," *Journal of Aircraft*, Vol. 27, No. 5, 1990, pp. 436–443.

[15]Batina, J. T., "Implicit Flux-Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Dynamic Meshes," AIAA 31st Structures, Structural Dynamics and Materials Conference, AIAA Paper 90-0936, Long Beach, CA, April 1990.

[16]Batina, J. T., "Accuracy of an Unstructured-Grid Upwind-Euler Algorithm for the ONERA M6 Wing," Accuracy of Unstructured Grid Techniques Workshop, NASA Langley Research Center, Hampton, VA, Jan. 16–17, 1990.

[17]Batina, J. T., "Three-Dimensional Flux-Split Euler Schemes Involving Unstructured Dynamic Meshes," AIAA 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference, AIAA Paper 90-1649, Seattle, WA, June 1990.

[18]Van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, 1982, pp. 507–512.

[19]Holmes, D. G., and Connell, S. D., "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA Paper 89-1392, June 1989.

[20]Pulliam, T. H., and Barton, J. T., "Euler Computations of AGARD Working Group Test Cases," AIAA 23rd Aerospace Sciences Meeting, AIAA Paper 85-0018, Reno, NV, Jan. 1985.

[21]Salas, M. D., (ed.), "Accuracy of Unstructured Grid Techniques Workshop," *NASA Conference Proceedings in Progress*, NASA Langley Research Center, Hampton, VA, Jan. 1990.

[22]Bland, S. R. (compiler), "AGARD Two-Dimensional Aeroelastic Configurations," AGARD-AR-156, Aug. 1979.

[23]Landon, R. H., "Oscillating and Transient Pitching, Data Set 3 in AGARD-R-702," *Compendium of Unsteady Aerodynamic Measurements*, NACA 0012, Aug. 1982.