Adaptive Quadrilateral and Triangular Finite-Element Scheme for Compressible Flows

R. Ramakrishnan*
Old Dominion University, Norfolk, Virginia
Kim S. Bey†
NASA Langley Research Center, Hampton, Virginia
and
Earl A. Thornton‡
Old Dominion University, Norfolk, Virginia

The development of an adaptive mesh refinement procedure for analyzing high-speed compressible flows using the finite-element method is described. This new adaptation procedure, which uses both quadrilateral and triangular elements, was implemented with two explicit finite-element algorithms—the two-step Taylor-Galerkin and the multistep Galerkin-Runge-Kutta schemes. A von Neumann stability analysis and a rotating "cosine hill" problem demonstrate the instability of the Taylor-Galerkin scheme when coupled with the adaptation procedure. For the same adaptive refinement scheme, the Galerkin-Runge-Kutta procedure yields stable solutions within its explicit stability limit. The utility of this new adaption procedure for the prediction of compressible flow features is illustrated for inviscid problems involving strong shock interactions at hypersonic speeds.

Nomenclature

A = element area
a, b = advection speeds, Eqs. (18) and (25)
$c^1,...,c^4$ = Runge-Kutta constants
d_1, d_2 = artificial dissipation fluxes
E_1, E_2 = flux components
e = element error
h = characteristic element length
i = imaginary unit, $\sqrt{-1}$
k = wave number of Fourier component
l = degree of space of finite-element solutions
L = length of element side
l_1, l_2 = components of unit normal vector
M = Mach number
$[M]$ = mass matrix
m = number of elements sharing a node
$[N]$ = element interpolation functions
n = number of element edges at a node
p = pressure
R = nodal residual
S = solution domain boundary
t = time
U = typical conservation variable
u = advection speed
x_1, x_2 = coordinate directions
ρ = key variable for refinement
α = threshold value for refinement
υ = Courant number
γ = ratio of specific heats
μ_2, μ_4 = dissipation constants
Ω = domain of interest
Δt = time step
$\Delta x, \Delta y$ = grid spacings in coordinate directions

Subscripts
i = summation index
j = index for nodes
s = constant surface quantity
o = stagnation values

Superscripts
k = index for multistep scheme
n = time step

Introduction

The use of computational fluid dynamics (CFD) for predicting compressible flow features has seen a dramatic increase in the last few years. The role of CFD has received a boost with recent initiatives to develop the hypersonic airplane, the "Orient Express." The aerospace plane, research vehicle X-30, is envisaged to have applications up to Mach 25. At these high speeds, the deformations and stresses that result from aerodynamic heating are significant and have a critical role in the aircraft design.

Research is under way at the NASA Langley Research Center to develop finite-element methods that can predict heating loads accurately for complicated flow configurations. The use of finite-difference methods for such flow problems is complicated by the need for complex meshing and the need for an effective way to blend meshes. Finite-element methods are characterized by their ease in handling completely unstructured meshes and their ability to include mesh refinement procedures with relative ease. Explicit finite-element schemes also have demonstrated their capacity to produce good results for a variety of flow situations and configurations.[1-4]

Received Jan. 7, 1988; presented as Paper 88-0033 at the AIAA 26th Aerospace Sciences Meeting, Reno, NV, Jan. 11–14, 1988; revision received Oct. 6, 1988.

*Research Associate, Department of Mechanical Engineering and Mechanics. Member AIAA.

†Aerospace Technologist, Aerothermal Loads Branch, Structural Mechanics Division.

‡Professor, Department of Mechanical Engineering and Mechanics. Associate Fellow AIAA.

Compressible flow situations contain discontinuities such as shocks and shear layers as well as regions of high gradients such as boundary layers that need to be adequately resolved. Since these flow phenomena occur over small-distance scales and may propagate as time evolves, the finite-element mesh needs to be adapted to model such flow situations accurately. Adaptive mesh refinement procedures can be used to effectively resolve interesting details in the flow region. The purpose of this paper is to describe the development of an adaptive refinement procedure to predict inviscid compressible flow features.

Adaptive mesh refinement procedures for compressible high-speed flows are of recent origin. Mesh refinement procedures for triangular finite-element meshes were initially detailed by Zienkiewicz, Lohner, and Morgan,[5] and the application of these procedures to steady[6] and transient[7] compressible flow problems has been extensively demonstrated. Adaptive procedures for finite-element meshes with quadrilateral elements have been developed by Oden et al.[8] and by Shapiro and Murman.[9]

Two popular methods of mesh adaptation for finite-element analysis of compressible flow are mesh enrichment and mesh movement. Mesh enrichment consists of adding degrees of freedom in regions where solution gradients are high. Mesh movement schemes, on the other hand, hold the degrees of freedom in a mesh constant but move the mesh to capture regions of high gradients. Mesh movement procedures generally lack the flexibility of enrichment procedures[10] and have a limited role in high-speed compressible flow computations. A mesh regeneration scheme developed by Peraire et al.[11] combines the advantages of enrichment and mesh movement schemes and is attractive for generating meshes with triangular elements.

The intent of this paper is to describe a refinement procedure based on quadrilateral elements that uses triangles to transition from crude to fine elements. The advantage of this approach over an enrichment approach using triangles is that significantly fewer elements are required. Thus computer storage requirements and processing time for quad-dominant meshes are smaller than meshes based exclusively on triangles. When the approach is extended to three dimensions, even higher savings will be realized.

Mesh Adaptation

Mesh Enrichment Schemes

The classical mesh refinement scheme in the context of the finite-element method is the addition of elements in regions of high gradients. Elements that lie in these regions are divided into smaller ones by a subdivision process. Both triangular and quadrilateral elements can be enriched by adding a central node and/or midside nodes. For a typical quadrilateral element, such a subdivision can result in the generation of four smaller elements with the possibility of the presence of midside nodes being unconnected to neighboring elements. These midside nodes are sometimes called "hanging nodes". The usual procedure to handle hanging nodes is to employ constraint equations that average nodal values of adjacent nodes to obtain the variables at the hanging node. One way to avoid these unconnected nodes is to transition from a crude quad mesh to a fine one using triangular elements. The advantage of this type of refinement is that no special constraint equations are needed, and the procedure provides a "natural" way to transition from crude to fine elements.

Refinement Indicators

The rationale for using refinement indicators is that, although it is possible to predict the location and strength of shocks, boundary layers, etc. for some simple flow situations, the analyst in general will not have prior knowledge of the location of regions containing sharp changes in flow variables. The decision to refine a particular region of the mesh can be based on either a priori or a posteriori error estimates. A posteriori error estimates are generally regarded as being computationally more efficient than a priori strategies.[8] The procedure adopted in this paper is to complete an analysis on a given mesh and then refine the mesh at a certain stage in the analysis. The refinement at this stage is based on error indicators computed on the initial mesh. The new mesh is used for the subsequent analysis, and then further refinements are performed if needed.

The aim of adaptive mesh refinement procedures is the minimization of maximum errors that occur in the finite-element domain. An optimal mesh satisfies the condition that the error is constant for all elements of the mesh. The adaptive mesh refinement procedure is designed to add elements in regions of high errors and to remove elements in regions of low errors, thereby equally distributing errors throughout the mesh. Since the exact solution is not known, the error is estimated from the finite-element solution. If ρ is considered as the key variable representative of solution behavior, then an error estimate for a typical element is computed from

$$e = \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial}{\partial x}\left(\frac{\partial \rho}{\partial y}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \rho}{\partial x}\right) + \frac{\partial^2 \rho}{\partial y^2} \tag{1}$$

For single shocks, the error estimate given in Eq. (1) is satisfactory; however, for multiple shock, computations show the need for a modified error estimate. To ensure that shocks of different strengths are adequately resolved and to prevent the mesh from being dominated by stronger shocks, derivatives in Eq. (1) are normalized by the gradient across the discontinuity. A typical derivative e_{xy} is thus computed as

$$e_{xy} = \frac{\dfrac{\partial}{\partial y}\left(\dfrac{\partial \rho}{\partial x}\right)}{\left|\dfrac{\partial}{\partial y}\left\{\left|\dfrac{\partial \rho}{\partial x}\right| + \epsilon \sum_{i=1}^{nd} \left|\dfrac{\partial \rho}{\partial x}\right|_i\right\}\right|} \tag{2}$$

where nd is the number of nodes in element i, and ϵ is a constant used to smooth out the error indicator in regions of oscillations. The adaptive refinement procedure refines all elements that satisfy the criterion

$$e > \alpha Y$$

and defines all elements that satisfy

$$e > \beta Y$$

where α and β are preset threshold constants, Y is the maximum element error over the entire domain. The key variable used is typically density, and α and β are usually 0.15 and 0.1, respectively.

The error estimates of Eqs. (1) and (2) require calculation of second derivatives of the key variables. For a typical element, a dependent variable within an element is interpolated from nodal values by

$$\rho(x, y) = [N(x, y)]\{\rho\} \tag{3}$$

where for a triangle the interpolation functions are linear, and for a quad the interpolation functions are bilinear. From Eq. (3), first derivatives can be computed, but the second derivatives needed for the error estimate, Eq. (1), cannot be computed. As an alternate approach, the following procedure is used. From Eq. (3), element first derivatives $\partial \rho^e / \partial x$ are computed by direct differentiation. Values of the first derivatives are then computed at nodes by assembling system equations of the form

$$\int_{\Omega}\{N\}[N]\,\mathrm{d}\Omega\left\{\frac{\partial\rho}{\partial x}\right\}=\int_{\Omega}\{N\}\,\mathrm{d}\Omega\,\frac{\partial\rho^{e}}{\partial x} \tag{4}$$

The coefficient matrix on the left-hand side of Eq. (4) is diagonalized to yield equations that are solved explicitly for the first derivatives. The procedure may be interpreted as computing a nodal derivative as a weighted average of the derivatives from the elements surrounding this nodes. Element area factors serve as weighting factors. The computation of the second derivatives follows the same steps. Element second derivatives are computed from nodal first derivatives by

$$\frac{\partial^{2}\rho^{e}}{\partial x^{2}}=\left[\frac{\partial N}{\partial x}\right]\left\{\frac{\partial\rho^{e}}{\partial x}\right\} \tag{5}$$

and finally nodal second derivatives are computed in a manner similar to that described by Eq. (4). The procedure lacks mathematical rigor, but it has proven accurate enough for computing error estimates.

Adaption Procedure

The starting point of the adaption procedure is a geometric or "skeleton" mesh. This mesh contains only a few elements and serves as the base mesh. The skeleton mesh, consisting of quadrilateral elements, needs to be completely structured to begin the adaption procedure. A structured mesh implies that each interior node in the domain is surrounded by the same number of elements. Before the first analysis, the mesh refinement program does an overall refinement where each element of the skeleton mesh is subdivided into four quad elements. This procedure is repeated a few times until the nodal density of the resulting mesh is deemed sufficient to obtain a solution that captures the main details of the flowfield.

Refinement indicators are computed based on the solution obtained on this "crude" mesh, and elements that need to be refined or derefined are identified. All elements in the mesh that have indicators above the preset refinement threshold value are enriched, whereas those elements that have values below the threshold derefinement value are coarsened. Elements with refinement indicators between the threshold values are left alone.

The refinement strategy used is such that at each mesh change, only one level of refinement or derefinement is permitted. On refinement of a typical element, the "subelements" that accrue could be all quads, or a combination of quads and triangles. The number and type of the resulting subelements depend on the refinement level of elements that surround this element.

Figure 1 shows the elements that result in a typical refinement and coarsening procedure. Figure 1a shows the original mesh where elements *B*, *C*, and *D* are to be refined. The mesh that results is seen in Fig. 1b. If on this refined mesh, element group *C*, which includes subelements *C*1, *C*2, *C*3, and *C*4, needs to be coarsened, the mesh that results after derefinement appears in Fig. 1c.

Finite-Element Algorithms

Two explicit-solution algorithms are investigated for use with the adaptive refinement scheme. The two-step Taylor-Galerkin algorithm has been used in the prediction of inviscid and viscous compressible flows for two- and three-dimensional problems.[2-4] Both triangular and quadrilateral elements have been used with this algorithm and have produced solutions of good quality. The multistep Galerkin-Runge-Kutta procedure also has been used for prediction of inviscid compressible flows,[6,9] although its application for viscous flows has been rather limited. Both algorithms use the method of weighted residuals to develop the finite-element matrices for spatial discretization but differ in the procedure used for time-marching to steady state.

Consider the development of solution algorithms for a system of conservation laws written in general form as

$$\boldsymbol{U}_{,t}+\boldsymbol{E}_{i,i}=0 \tag{6}$$

where the vector of variables $\boldsymbol{U}$ and the flux vector $\boldsymbol{E}_i$ may be defined to model the advection equation, Euler equations, or the Navier-Stokes equations. The coordinate directions are denoted by x_i, and the domain can be discretized using quads, triangles, or a combination of both element types.

Taylor-Galerkin Algorithm

The two-step Taylor-Galerkin finite-element formulation uses a Taylor-series expansion in time to establish the recurrence relations for time marching, and a weighted residual formulation describes the spatial distribution of the nodal variables.

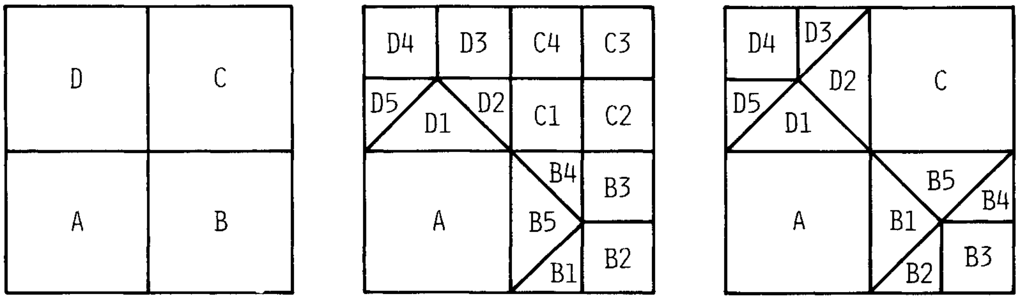
At the first step or time level, flow variables within an element are assumed constant. The value of a typical variable U at the half-step $U^{n+1/2}$ is computed from

$$AU^{n+1/2}=\int_{A}[N]\,\mathrm{d}A\{U\}^{n}-\frac{\Delta t}{2}\int_{A}[N_{,i}]\,\mathrm{d}A\,\{E_i\}^{n} \tag{7}$$

where $\{E_i\}$ are the flux vectors.

At the second time level t^{n+1}, the new values of the nodal variables U^{n+1} are computed from

$$\begin{aligned}[M]\{U\}^{n+1}&=[M]\{U\}+\Delta t\int_{A}[N_{,i}]\,\mathrm{d}AE_{i}^{n+1/2}\\&\quad-\Delta t\int_{S}[l_{i}E_{is}^{n+1/2}]\{N\}\,\mathrm{d}S\end{aligned} \tag{8}$$



a) Original mesh

b) Refinement of elements *B*, *C*, and *D*

c) Derefinement of element *C*

Fig. 1 Example of adaptation procedure using triangular transition elements.

where $[M]$ denotes the element mass matrix

$$[M] = \int_A \{N\}[N]\,\mathrm{d}A \tag{9}$$

The last term in Eq. (8) represents boundary fluxes on the outflow surface S where l_i are direction cosines of a unit vector normal to the surface. The algorithm is implemented in an explicit manner with the mass matrix $[M]$ replaced by the lumped, diagonal mass matrix. Complete details of the Taylor-Galerkin finite-element formulation appear in Refs. 2–4.

Galerkin-Runge-Kutta Algorithm

The system of conservation laws shown in Eq. (6) can be written as

$$\boldsymbol{U}_{,t} = -\boldsymbol{E}_{i,\,i} \tag{10}$$

Application of the method of weighted residuals results in the finite-element equations

$$\int_A [N]\left\{\frac{\partial \boldsymbol{U}}{\partial t}\right\}\mathrm{d}A = -\int_A [N](\mathbf{E}_{i,\,i})\,\mathrm{d}A \tag{11}$$

where $[N]$ is the element interpolation function and A the domain of interest. Integrating by parts on the right-hand side terms yields

$$[M]\left\{\frac{\partial \boldsymbol{U}}{\partial t}\right\} = \int_A \{N_{,i}\}[N]\,\mathrm{d}A\{\boldsymbol{E}_i\} - \int_S \{N\}[N]\,\mathrm{d}S\,[l_i\boldsymbol{E}_{is}] \tag{12}$$

where the fluxes $\boldsymbol{E}_i$ are interpolated the same way as the conservation variables.

The time marching scheme is similar to the multistep time integration scheme that appears in Ref. 9 and can be written as

$$\boldsymbol{U}_j^{(k)} = \boldsymbol{U}_j^n + c^k\left[\frac{\Delta t_j}{\boldsymbol{M}_{jj}} R_j(\boldsymbol{U}^{k-1}) + D\right], \qquad k = 1,..,4 \tag{13a}$$

$$\boldsymbol{U}_j^{n+1} = \boldsymbol{U}_j^{(4)} \tag{13b}$$

where the dissipation operator D can be written as

$$D = D_2^n + D_4^n \tag{14}$$

and D_2 and D_4 are second- and fourth-difference dissipation terms that may be "frozen" for each time step.

The second-difference artificial dissipation term is needed to stabilize solutions in the presence of sharp gradients. This dissipation operator is given by

$$D_2 = d_{i,\,i}$$

$$d_i = k_i\boldsymbol{U}_{,i}, \qquad (i \text{ not summed}) \tag{15}$$

where k_i control the amount of dissipation added and are given by

$$k_i = \frac{\mu_2 h^4}{4p}|p_{,ii}|, \qquad (i \text{ not summed}) \tag{16}$$

where μ_2 is the second-difference dissipation constant and p the pressure. The first derivatives of pressure are calculated as element quantities, and nodal second derivatives are then obtained from the relation

$$\int_\Omega \{N\}[N]\,\mathrm{d}\Omega\left\{\frac{\partial^2 p}{\partial x^2}\right\} = -\int_\Omega \{N_{,x}\}\,\mathrm{d}\Omega\,\frac{\partial p^e}{\partial x}$$

where the comma implies differentiation.

The linear fourth-difference operator is needed to damp out spurious oscillations and provide background dissipation and is given by

$$D_4 = \mu_4(\Delta x_i)^4\boldsymbol{U}_{,iiii} \tag{17}$$

The fourth differences are evaluated by repeated application of a 9-point Laplacian stencil. For a typical node i, second differences are evaluated by considering the elements that contribute to this node as given by

$$D_2 = \sum_{k=1}^{n} U_K^e - 4m\boldsymbol{U}_i$$

where m is the number of elements that share node i and

$$\boldsymbol{U}_K^e = \sum_{i=1}^{nd} \boldsymbol{U}_i$$

This differencing scheme differs from that described for unstructured triangular meshes,[12] which compute second differences from

$$D_2 = \sum_{k=1}^{n} \boldsymbol{U}_k - n\boldsymbol{U}_i$$

where n is the number of edges meeting at node i, and differences are taken between the two nodes at either end of each edge. For finite-element meshes generated by the adaptation procedure, the use of either dissipation scheme results in solutions of good accuracy. Computational experience also indicates that for completely arbitrary meshes the dissipation scheme suggested in Ref. 12 is to be preferred.

Von Neumann Stability Analysis

The stability analysis is performed for the two-step Taylor-Galerkin and the four-step Galerkin-Runge-Kutta algorithms applied to the simple advection equation in one and two dimensions.

One-Dimensional Advection

For the one-dimensional advection equation, $\boldsymbol{U} = \boldsymbol{u}$ and $\boldsymbol{E} = \boldsymbol{au}$, and Eq. (6) reduces to

$$u_{,t} + au_{,x} = 0 \tag{18}$$

with

$$u(x,o) = f(x) \tag{19}$$

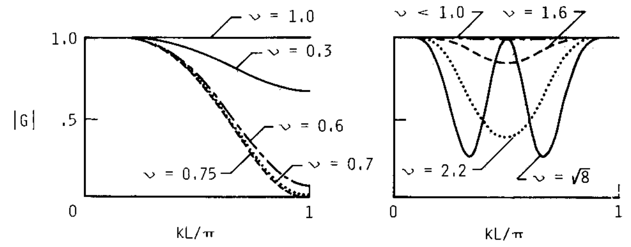
where a is the speed of wave propagation, and $f(x)$ is the prescribed initial distribution. Equation (18) has the exact solution given by

$$u(x,t) = f(x - at) \tag{20}$$

The stability analysis is facilitated by writing the Taylor-Galerkin and Galerkin-Runge-Kutta schemes for the one-dimensional advection equation in finite-difference form. The notation $u_j{}^{n+1}$ is used to represent the solution to the discretized Eq. (18) at location $x_j = j\Delta x$ and at time $t^{n+1} = (n+1)\Delta t$. The finite-difference form that results for the two-step Taylor-Galerkin algorithm is identical to the two-step Lax-Wendroff finite-difference scheme

$$u_j^{n+1} = u_j^n - \frac{\upsilon}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{\upsilon^2}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{21}$$

where $\upsilon = a\Delta t/\Delta x$ is the Courant number.



a) Taylor-Galerkin **b) Runge-Kutta**

Fig. 2 Comparison of amplification factors for Taylor-Galerkin and Runge-Kutta algorithms.

The finite-difference equation for the four-step Galerkin-Runge-Kutta scheme for the time-step Δt is

$$\begin{aligned} u_j^{n+1} &= u_j^n - \frac{\upsilon}{2}(u_{j+1}^n - u_{j-1}^n) \\ &+ \frac{\upsilon^2}{8}(u_{j+2}^n - 2u_j^n + u_{j-2}^n) \\ &- \frac{\upsilon^3}{48}(u_{j+3}^n - 3u_{j+1}^n \\ &+ 3u_{j-1}^n - u_{j-3}^n) + \frac{\upsilon^4}{384}(u_{j+4}^n - 4u_{j+2}^n \\ &+ 6u_j^n - 4u_{j-2}^n + u_{j-4}^n) \end{aligned} \tag{22}$$

The amplification factor $G = u_j^{n+1}/u_j^n$ for a numerical scheme is a measure of the growth of error components from one time step to the next. By substituting the difference form of the solution, expressions for the amplification factor G are obtained for the two algorithms. For the two-step Taylor-Galerkin algorithm, the amplification factor is given by

$$G = 1 + \upsilon^2(\cos\theta - 1) - i\upsilon\sin\theta \tag{23}$$

where $\theta = k\Delta x$, and k is the wave number of the Fourier component. For the Runge-Kutta formulation,

$$\begin{aligned} G &= (1 - \upsilon^2/8)^2 + (\upsilon^2/4)(1 - \upsilon^2/12)\cos 2\theta + (\upsilon^2/192)\cos 4\theta \\ &- i[\upsilon(1 - \upsilon^2/8)\sin\theta + (\upsilon^3/24)\sin 3\theta] \end{aligned} \tag{24}$$

The numerical stability requirement for both algorithms is that the modulus of G remain bounded by unity. This leads to the following limits on the Courant number:

Taylor-Galerkin $|\upsilon| < 1$

Runge-Kutta $|\upsilon| < \sqrt{8}$

A comparison of the relative errors in amplitude for the two algorithms appears in Fig. 2. The Taylor-Galerkin algorithm is seen to be highly dissipative (Fig. 2a) for Courant numbers less than 1, whereas the Runge-Kutta algorithm is dissipative only for selected wave numbers.

Two-Dimensional Advection

The behavior of the two-step Taylor-Galerkin and the four-step Runge-Kutta algorithms in the presence of different element types is studied using the two-dimensional equation

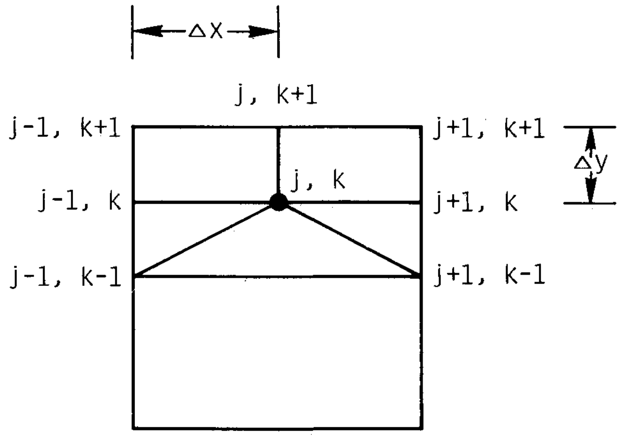


Fig. 3 Typical interior node of a mesh with triangular transition elements.

$$u_{,t} + au_{,x} + bu_{,y} = 0 \tag{25}$$

where a and b are advection speeds in the coordinate directions. It is instructive to look at the numerical response of the two-step Taylor-Galerkin scheme for meshes that contain 1) all quads, 2) all triangles, and 3) a combination of triangles and quads that results from the adaptive refinement strategy.

The amplification factor for an interior node of a typical mesh containing all quad elements is given by

$$\begin{aligned} G &= 1 - 2\upsilon_x^2\sin^2(\theta_x/2)\cos^2(\theta_y/2) \\ &- 2\upsilon_y^2\sin^2(\theta_y/2)\cos^2(\theta_x/2) \\ &- \upsilon_x\upsilon_y\sin\theta_x\sin\theta_y - i[\upsilon_x\sin\theta_x\cos^2(\theta_y/2) \\ &+ \upsilon_y\sin\theta_y\cos^2(\theta_x/2)] \end{aligned} \tag{26}$$

where $\theta_x = k_x\Delta x$ and $\theta_y = k_y\Delta y$. Here k_x and k_y are the wave numbers of the Fourier components, υ_x and υ_y the Courant numbers in the coordinate directions, and Δx and Δy are the mesh spacings.

The amplification factor for a typical interior node of a triangular element mesh depends on the orientation of the triangles. One choice is to bisect each quadrilateral by the same diagonal. Note that only the six triangles that surround the interior node contribute to the equation for the node. The amplification factor for an interior node of the triangular element mesh is given by

$$\begin{aligned} G &= 1 - 2\upsilon_x^2\sin^2(\theta_x/2) - 2\upsilon_y^2\sin^2(\theta_y/2) \\ &+ \upsilon_x\upsilon_y[\cos\theta_x + \cos\theta_y - \cos(\theta_x - \theta_y) - 1] \\ &- i\{(\upsilon_x/3)[\sin\theta_y + 2\sin\theta_x + \sin(\theta_x - \theta_y)] \\ &+ (\upsilon_y/3)[\sin\theta_x + 2\sin\theta_y - \sin(\theta_x - \theta_y)]\} \end{aligned} \tag{27}$$

To determine the stability limit for the Courant number, consider a mesh with $\Delta x = \Delta y$ and waves with $k_x = k_y$ and thus $\upsilon_x = \upsilon_y = \upsilon$. This leads to the stability criteria that for a quadrilateral element mesh $\upsilon < 1$ and for a corresponding triangular element mesh $\upsilon < 0.5$.

A typical node that results from the adaptive refinement strategy of transitioning from crude to fine quad meshes using triangles is shown in Fig. 3. The amplification factor for such a node is given by

$$
\begin{aligned}
G = 1 &- (\upsilon_x/14)\sin\theta_x\sin\theta_y + (\upsilon_y/12)\\
&\times[-3(\cos\theta_x+\cos\theta_y) + 5\cos\theta_x\cos\theta_y + 1]\\
&+ (3\upsilon_x^2/14)(\cos\theta_x - 1)(3+\cos\theta_y) + (3\upsilon_y^2/14)\\
&\times[3(\cos\theta_x\cos\theta_y - 1) - \cos\theta_x + \cos\theta_y]\\
&- (3\upsilon_x\upsilon_y/7)\sin\theta_x\sin\theta_y + i[-\upsilon_x/2\sin\theta_x(1+\cos\theta_y)\\
&- (\upsilon_y/12)\sin\theta_y(3+11\cos\theta_x)\\
&+ (3\upsilon_x^2/14)(\cos\theta_x - 1)\sin\theta_y + (3\upsilon_y^2/14)\\
&\times\sin\theta_y(1-\cos\theta_x) + (3\upsilon_x\upsilon_y/7)\sin\theta_x(1-\cos\theta_y)]
\end{aligned}
\tag{28}
$$

The von Neumann analysis shows that this node is unstable for all reasonable Courant numbers. This instability is demonstrated numerically for the rotating cosine hill problem, i.e., by rotating a cone through the mesh shown in Fig. 4. The cone initially starts in the crude region of the mesh and is rotated counterclockwise. A time history of a typical transition node shows the growth of solution errors clearly after one complete revolution, and the error growth continues with succeeding revolutions (Fig. 5). The same numerical experiment shows the Galerkin-Runge-Kutta procedure (without any explicit dissipation) to be stable. Based on these results, the two-step Taylor-Galerkin algorithm has been abandoned for meshes of quads and triangles with transition nodes of the type shown in Fig. 3. Only the Runge-Kutta time marching algorithm is used for subsequent computations with the adaptive refinement scheme.

Computational Results

To validate the Galerkin-Runge-Kutta finite-element approach and to demonstrate the capability of the adaption procedure to converge on inviscid compressible flow features, two problems of current interest to the Aerothermal Loads Branch at NASA Langley are analyzed. The adaptive finite-element procedure was implemented on the Langley VPS-32, and computational experience with the program indicates processing rates to average about 5.E-05 CPU/time step/node. Comparison solutions obtained by empirical and experimental studies are also presented to gauge solution accuracy.

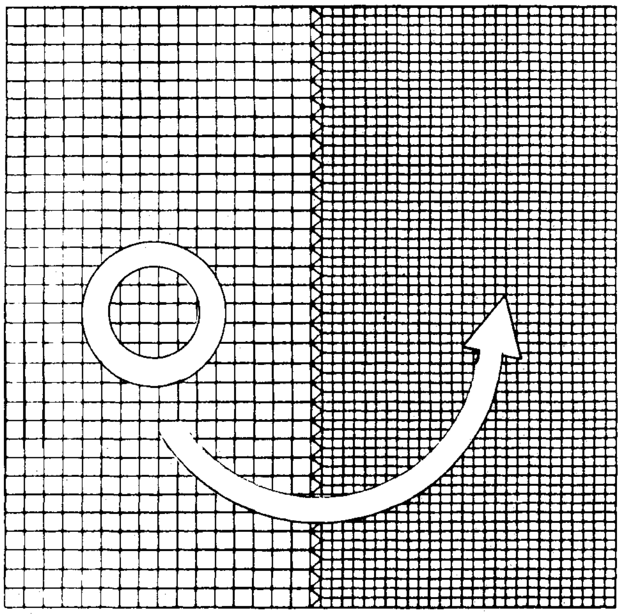


Fig. 4 Mesh with triangular transition elements used to study stability of algorithms.

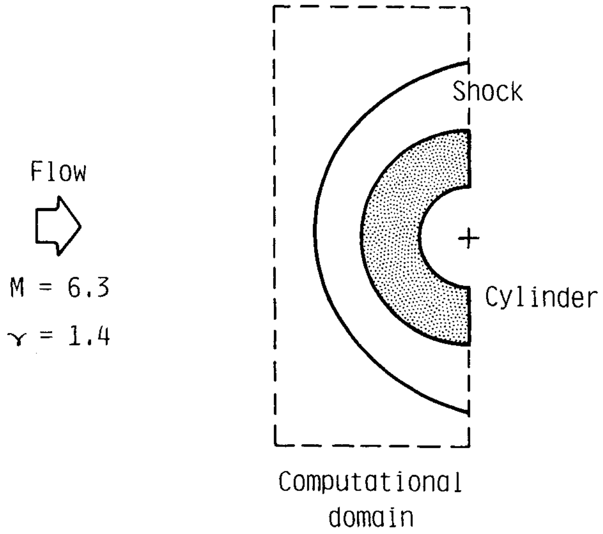


Fig. 6 Schematic of undisturbed flow over cowl leading edge.

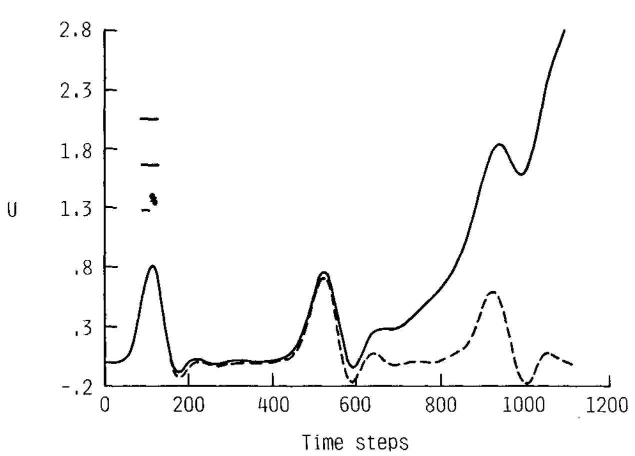
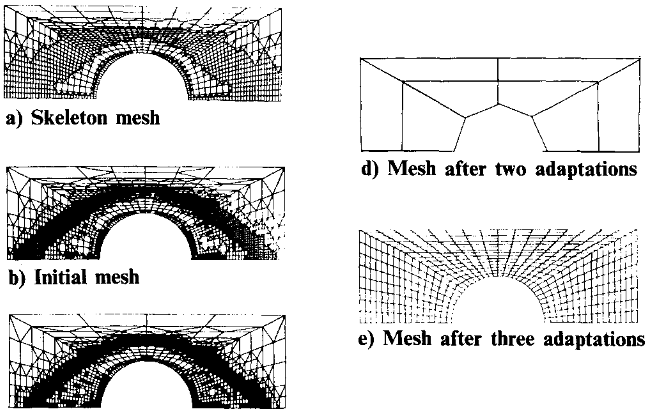


Fig. 5 Comparison of time histories for transition node using Taylor-Galerkin and Galerkin-Runge-Kutta algorithms.



a) Skeleton mesh

b) Initial mesh

c) Mesh after one adaptation

d) Mesh after two adaptations

e) Mesh after three adaptations

Fig. 7 Skeleton mesh, initial mesh, and meshes after successive adaptations for undisturbed flow over cowl leading edge.

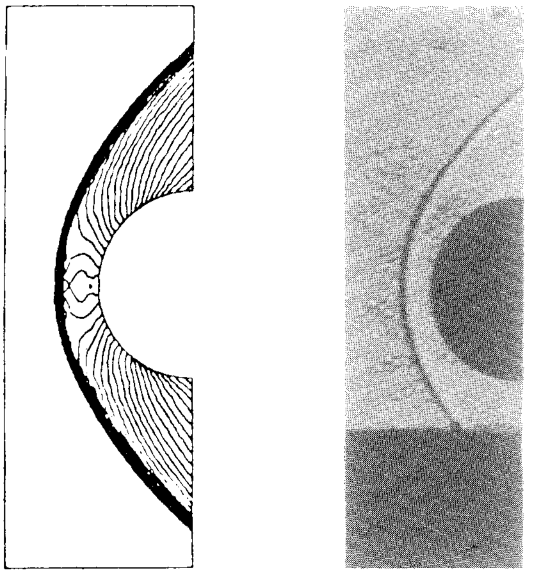


Fig. 8 Comparison of density contours for adapted mesh with Schlieren photograph.

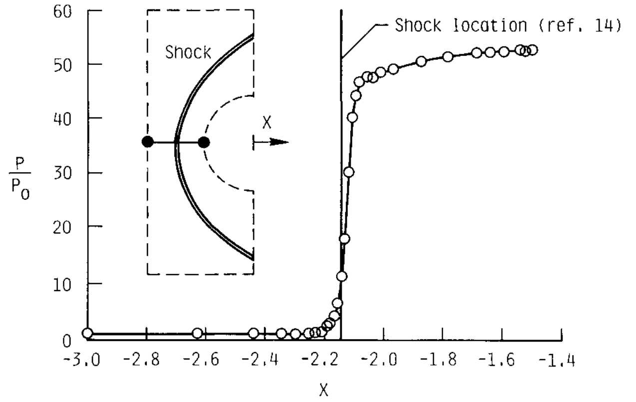


Fig. 9 Distribution of pressure along the centerline for undisturbed flow over cowl leading edge.

Undisturbed Flow Over a Cylindrical Cowl Leading Edge

The first problem presented is the Mach 6.3 flow over an undisturbed cylindrical cowl leading edge, the configuration of which is shown in Fig. 6. The flowfield is characterized by the bow shock that stands off the leading edge with the flow expanding over the body of the cylinder. The skeleton mesh, the initial mesh used in the analysis, and the succession of meshes that result from the adaption procedure appear in Fig. 7. It is seen that the number of triangles in the mesh increases with adaption, with more and more transition elements being needed at each refinement level. Convergence at each refinement level is assumed when the L_2 norm of the change in conservation variables drops over three orders of magnitude. Typically, around 1500 iterations are required at each refinement level to satisfy this convergence criterion. The pressure force on the cylinder surface is computed at each refinement level, and the analysis is terminated when successive refinements indicate minimal change in the computed pressure force. The mesh obtained after three refinements (shown on Fig. 7e) contains 5955 nodes and 5973 elements, and the density contours obtained on this mesh are compared with the Schlieren photograph[13] in Fig. 8. The results obtained indicate good resolution of the bow shock and the flow expansion over the body of the cylinder.

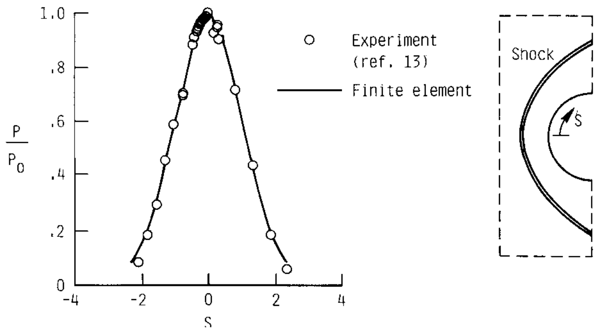


Fig. 10 Comparison of pressure distribution along surface of cowl leading edge.

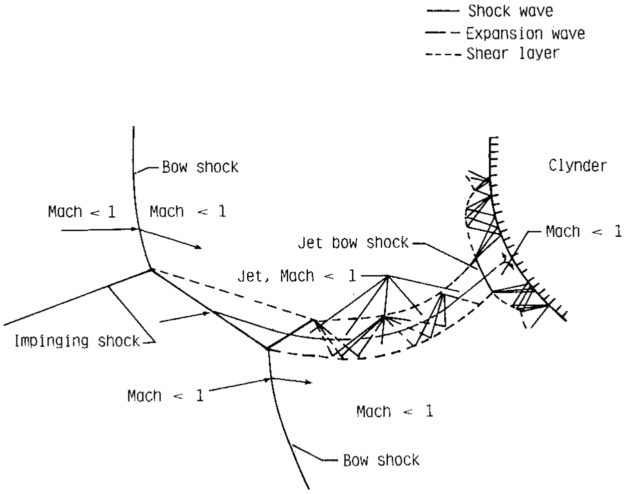


Fig. 11 Flow features of type IV interference pattern.

Figure 9 shows the distribution of pressure along the centerline obtained by the finite-element analysis. The location of the bow shock is seen to compare well with the shock location predicted by Billig.[14] Figure 10 compares the distribution of pressure along the surface of the cylinder with experimental data.[13] The normalized pressure distribution is seen to be in good agreement with the experimental pressures.

Shock Impingement on Cowl Leading Edge

The second problem analyzed is the flowfield associated with the cowl lip of a hypersonic vehicle. The compression waves generated from the forebody of such a vehicle coalesce to form an oblique shock that impinges on the leading edge of the cowl. The shock interference pattern that results depends on the intersection point of the impinging oblique shock and the cylinder bow shock. An interference pattern of particular interest, described as type IV in Ref. 13, is shown in Fig. 11. The flowfield is characterized by the development of shear layers that contain a supersonic "jet" that is submerged within the subsonic region between the body and the bow shock wave. The jet terminates through a normal shock creating a small stagnation region as it impinges on the leading edge. The impingement of the jet on the body causes the pressure and heating rates at this point to be augmented several times the undisturbed values. The intent of the modeling procedure is to capture the complex inviscid flow features that arise in the flow and make comparisons with the experimental observations of Wieting and Holden.[15]

A schematic of the flow configuration (Fig. 12) shows the shock coming off the leading edge (simulating the vehicle forebody) interacting with the bow shock that forms ahead of the cylindrical leading edge. The inflow boundary used for the analysis is divided into two regions. Region ABC has conditions specified as freestream, while region CDE has inflow values that correspond to conditions downstream of an oblique shock. In the finite-element model, the impinging shock is always spread over one element at the inflow boundary.

The mesh obtained after five levels of adaption contains 7380 nodes and 7513 elements and is shown in Fig. 13. Note that shocks of different strengths—the weaker bow shock on the upper half of the mesh, the strong bow shock at the lower half, and the impinging shock—are all refined uniformly. The density contours that result on this final mesh are detailed in Fig. 14, and a comparison of the contours with the Schlieren photograph appears in Fig. 15. A close inspection of the density contours shows the presence of the shear layer, the oblique shocks that result from the interaction, and the supersonic jet that impinges on the surface of the body.

A comparison of the surface pressures on the cylinder using the finite-element approach with the experimental data of Ref. 15 appears in Fig. 16. The peak pressure on the cylinder surface compares well with the experimental pressure peak, although the location of the peak is seen to be shifted to the left. This discrepancy can be attributed, in large measure, to uncertainties in the location of the impinging shock on the inflow plane. Numerical[6] and experimental[13,15] studies indicate slight variations in the location of this shock having a significant effect on the flowfield and surface pressure distribution.

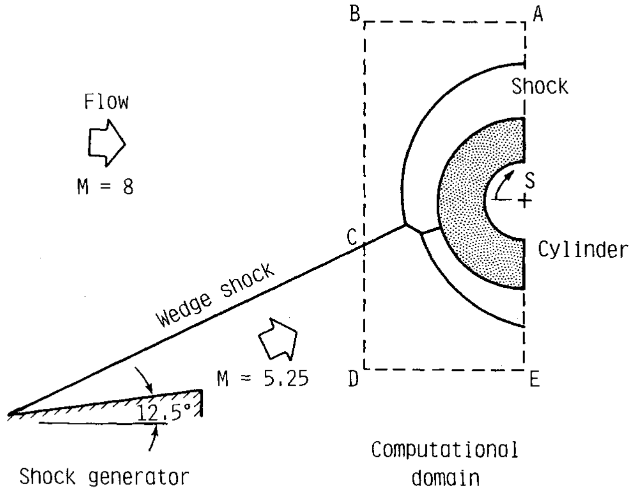


Fig. 12 Schematic of flowfield due to shock impingement on cowl leading edge.

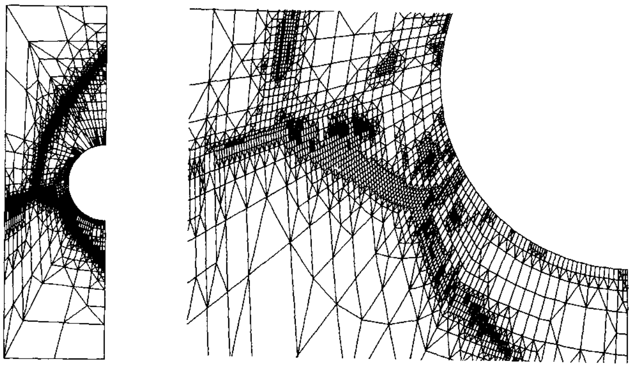


Fig. 13 Finite-element mesh for shock impingement on cowl leading edge.

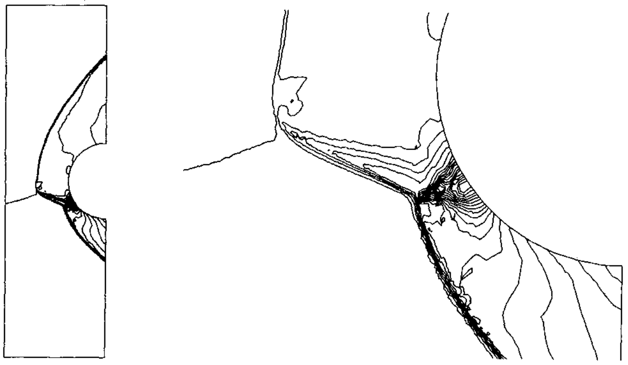


Fig. 14 Density contours for shock impingement on cowl leading edge.

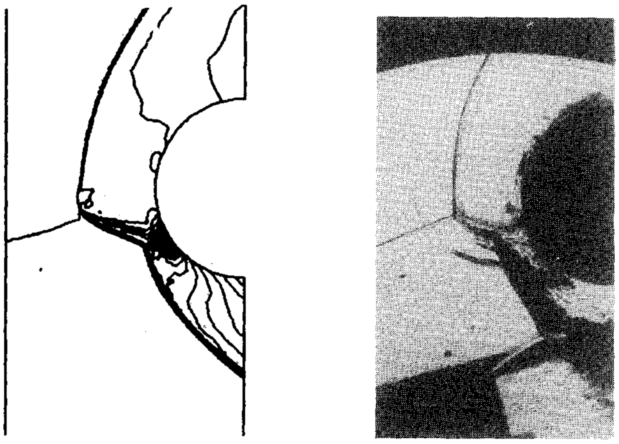


Fig. 15 Comparison of density contours with Schlieren photograph for shock impingement on cowl leading edge.

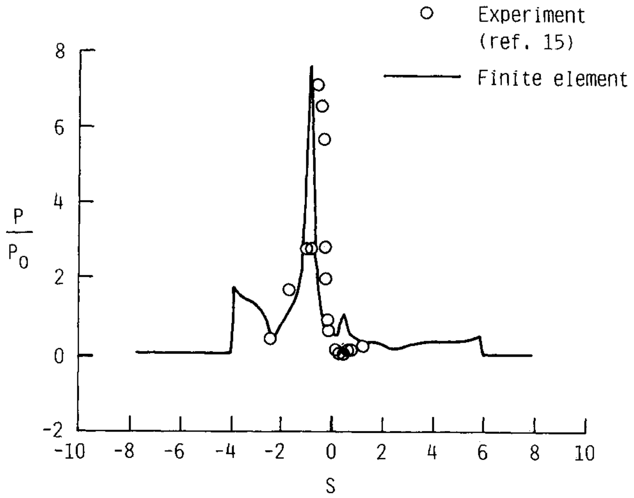


Fig. 16 Comparison of pressure distribution on surface of cowl leading edge.

Concluding Remarks

An adaptive refinement procedure that uses both triangular and quadrilateral elements is described. Triangular elements are used to transition from crude to fine quadrilateral elements. Basic details of the adaption procedure are illustrated, and the refinement indicators used for adaption are presented. The adaptation is based on refinement indicators that are designed to handle shocks of differing strengths.

Two finite-element algorithms are investigated for use with the adaptive refinement strategy. A von Neumann stability analysis and a rotating cosine hill advection problem lead to the following conclusions: 1) for the transition node that results from adaptation, the two-step Taylor-Galerkin is unstable, and 2) the Galerkin-Runge-Kutta procedure is stable at transition nodes.

Numerical results are presented for inviscid compressible flow analysis to assess the utility of the adaptation procedure with the four-step Galerkin-Runge-Kutta algorithm. Two problems of current interest, the hypersonic flow over a cowl leading edge, with and without shock impingement, are analyzed. The results obtained for these problems indicate the ability of the adapted four-step Galerkin-Runge-Kutta scheme to model shocks and discontinuities with good resolution.

Acknowledgment

The authors are pleased to acknowledge the continuing support of the NASA Langley Aerothermal Loads Branch and the advice and encouragement of Allan R. Wieting, our technical monitor. The research of R. Ramakrishnan and E. A. Thornton was supported by NASA Research Grant NSG-1321.

References

[1]Lohner, R., Morgan, K., and Zienkiewicz, O. C., "The Solution of Non-Linear Hyperbolic Equation Systems by the Finite Element Method," *International Journal of Numerical Methods in Fluids,* Vol. 4, 1984, pp. 1043-1063.

[2]Bey, K. S., Thornton, E. A., Dechaumphai, P., and Ramakrishnan, R., "A New Finite Element Approach for Prediction of Aerothermal Loads-Progress in Inviscid Flow Computations," AIAA Paper 85-1533, July 1985.

[3]Thornton, E. A., Ramakrishnan, R., and Dechaumphai, P., "A Finite Element Approach for Solution of the 3D Euler Equations," AIAA Paper 86-0106, Jan. 1986.

[4]Thornton, E. A., Dechaumphai, P., and Vermaganti, G., "A Finite Element Approach for Prediction of Aerothermal Loads," AIAA Paper 86-1050, May 1986.

[5]Zienkiewicz, O. C., Lohner, R., and Morgan, K., "High Speed Inviscid Compressible Flow by the Finite Element Method," Mathematics of Finite Elements and Applications, J. R. Whiteman, editor, Academic, 1985, pp. 1–26.

[6]Stewart, J. R., Thareja, R. R., Wieting, A. R., and Morgan, K., "Inviscid Finite Element Validations of Shock Interference on a Cylindrical Leading Edge," AIAA Paper 88-0368, Jan. 1988.

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

[8]Oden, J. T., Demkowicz, L., Strouboulis, T., and Devloo, P., "Adaptive Methods for Problems in Solid and Fluid Mechanics," *Adaptive Methods and Error Refinement in Finite Element Computation,* edited by I. Babuska, O. C. Zienkiewicz, J. P. de S. R. Gago, and A, de Oliveira, Wiley, London, 1986.

[9]Shapiro, R. A. and Murman, E. M., "Cartesian Grid Finite Element Solutions to the Euler Equations," AIAA Paper 87-0559, Jan. 1987.

[10]Lohner, R., Morgan, K., and Zienkiewicz, O. C., "Adaptive Grid Refinement for the Euler and Compressible Navier-Stokes Equations," *Adaptive Methods and Error Refinement in Finite Element Computation*, edited by I. Babuska, O. C. Zienkiewicz, J. P. de S. R. Gago, and A. de Oliveira, Wiley, London, 1986.

[11]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, 1987, pp. 449–466.

[12]Mavriplis, D. and Jameson, A., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes," AIAA Paper 87-0353, Jan. 1987.

[13]Wieting, A. R., "Experimental Study of Shock Wave Interference on a Cylindrical Leading Edge," Ph.D. Dissertation, Old Dominion Univ., Norfolk, VA, 1987.

[14]Billig, F. S., "Shock-Wave Shapes around Spherical- and Cylindrical-Nosed Bodies," *Journal of Spacecraft and Rockets*, Vol. 4, June 1967, pp. 822–823.

[15]Wieting, A. R. and Holden, M. S., "Experimental Study of Shock Wave Interference Heating on Cylindrical Leading Edge," AIAA Paper 87-1511, June 1987.