AIAA JOURNAL
Vol. 32, No. 2, February 1994

Upwind Method for Simulation of Viscous Flow on Adaptively Refined Meshes

M. J. Aftosmis*
U.S. Air Force Wright Laboratory, Wright-Patterson Air Force Base, Ohio 45433

A new node-based upwind scheme for the solution of the three-dimensional Navier-Stokes equations on adaptively refined meshes is presented. The method uses a second-order upwind total variation diminishing scheme to integrate the convective terms and discretizes the viscous terms with a new compact central difference technique. Grid adaptation is achieved through directional division of hexahedral cells in response to evolving features as the solution converges. The method is advanced in time with a multistage Runge-Kutta time stepping scheme. Two- and three-dimensional examples establish the accuracy of the inviscid and viscous discretization. These investigations highlight the ability of the method to produce crisp shocks, while accurately and economically resolving viscous layers. The representation of these and other structures is shown to be comparable to that obtained by structured methods. Further three-dimensional examples demonstrate the ability of the adaptive algorithm to effectively locate and resolve multiple scale features in complex three-dimensional flows with many interacting, viscous, and inviscid structures.

Nomenclature

e = total internal energy per unit mass
$\bar{\bar{F}}$ = complete flux density tensor
g = limiter function
$\bar{n}$ = unit normal vector $[n_x, n_y, n_z]^T$
p = local static pressure
Q_k = vector of numerical flux through face k
R_s = shock refinement parameter
$\mathfrak{R}$ = right eigenvalue matrix of flux Jacobian in transformed space
$\bar{S}_z$ = surface vector of face k, $[S_x, S_y, S_z]_k^T$
T = threshold for cell division
$\bar{U}$ = state vector $[\rho, \rho u, \rho v, \rho w, e]^T$
u, v, w = Cartesian velocity components
V_i = volume of cell i
δ = small parameter preventing zero wave speed, $\delta \ll 1$
λ = eigenvalue of flux Jacobian matrix
ρ = density
Φ_k = limited flux function at face k

I. Introduction

THE accurate simulation of flows with features spanning many length scales presents a major challenge to computational fluid dynamics. This issue becomes especially critical in three-dimensional flows over complex configurations, or even in flows over simple configurations at flow conditions where multiple, interacting three-dimensional features must be correctly represented.

The last decade has seen the emergence of two basic approaches to surmounting this fundamental difficulty. High resolution upwind schemes attempt to represent shocks and contact discontinuities with as few cells as possible,[1–6] whereas adaptive grid techniques automatically adjust the local mesh dimension in an attempt to resolve the scale of the local flow physics.[5–10] Therefore, the combination of a high resolution upwind algorithm together with unstructured mesh adaptation suggests a natural and extremely flexible platform for simulating flowfields with features spanning a variety of disparate length scales. The current work adopts this combined approach and uses adaptively refined hexahedral meshes to provide a foundation for a second order upwind algorithm based on the upwind total variation diminishing (TVD) technique of Harten and Yee.[11]

Mesh adaptation through directional cell division forms a flexible and efficient procedure for resolving finer length scales within the flowfield.[6,8,9] The technique avoids the limited nature of grid redistribution algorithms and minimizes computational expense since points are only added in regions of the simulation that require further enhancement. This flexibility requires a solver capable of dividing an arbitrary collection of cells and necessitates fully unstructured data storage. Despite the use of hexahedral meshes and often structured looking starting meshes, the solver must be free of assumptions about mesh topology and must treat any mesh as simply a collection of cells with some explicitly defined connectivity.

Although interest is currently growing in truly multidimensional[12] and multidirectional[13] upwind techniques, most successful and robust upwind techniques still rely on successive application of essentially one-dimensional operators through space operator splitting. On triangular or tetrahedral meshes, the implementation of higher order upwind operators (with their large discretization stencils) is not straightforward, and several approaches have been suggested in the recent literature.[14,15] The use of hexahedral cells, however, provides a clear mapping of this discretization stencil onto the domain, avoiding the ambiguities associated with higher order upwind stencils on triangular meshes.[6] Additionally, such meshes easily provide the high quality regular mesh near wall boundaries that is necessary for accurate evaluation of the viscous fluxes in the Navier-Stokes equations.

Nearly all unstructured techniques require an initial mesh to begin the numerical simulation. In the current method, this mesh typically begins as a single- or multiple-block structured mesh. Thus, a good quality background mesh must still be generated by conventional techniques, and the flexibility afforded by the Delaunay, advancing front, or other techniques available for generating tetrahedral meshes is not currently available for hexahedral meshes. However, this drawback is no more severe than it is for structured techniques and is largely offset by both the natural mapping of the upwind operator and the accuracy advantage offered by the use of hexahedral cells in viscous regions.

II. Description of Method

The numerical method relies on Harten and Yee's upwind TVD algorithm[11] to integrate the inviscid fluxes whereas the viscous terms use central difference modeling. This spatial discretization is

Received Oct. 30, 1992; revision received March 23, 1993; accepted for publication March 24, 1993.

*Numerical Aerodynamicist; currently stationed at the NASA Ames Research Center, MS T045-2, Moffett Field, CA 94039. Member AIAA.

advanced in time through a modified Runge-Kutta procedure. An option for central differencing of the inviscid fluxes also exists. This option is essentially a node-based version of the well-known central scheme by Jameson with blended second- and fourth-order artificial dissipation.[16] The central difference option was included primarily to permit direct comparison of discrete solutions from upwind and central difference simulations on identical adapted meshes.

A. Governing Equations

The Navier-Stokes equations describe the unsteady flow of a viscous gas and may be written in integral form for a general three-dimensional region V.

$$\iiint_V \frac{\partial}{\partial t}\bar{U}\,\mathrm{d}V = -\oint\!\!\oint_{\partial V} \bar{\bar{F}}\bullet\bar{n}\,\mathrm{d}s \tag{1}$$

$$\bar{\bar{F}} = \bar{\bar{F}}_I - \bar{\bar{F}}_V \tag{2}$$

Here, $\bar{U}$ is the state vector of conserved variables, and $\bar{\bar{F}}$ is the complete tensor of flux density which contains both viscous and inviscid components, and ∂V is the closed boundary of V with the outward facing unit normal vector $\bar{n} = [n_X, n_Y, n_Z]^T$. This set of equations is closed through the assumption of a calorically perfect gas and adopting the Stokes hypothesis. Assuming an isotropic, Newtonian fluid yields a symmetric shear stress tensor. Constant Prandtl number modeling is assumed throughout the domain, and Sutherland's law describes the variation of viscosity with temperature.

B. Spatial Integration of Convective Fluxes

The spatial discretization of the governing equations may be conveniently developed by focusing on the symbolic form of the equations given by expression (1). Considering a control volume fixed in time, and applying the mean value theorem, this equation may be re-cast as

$$\left(\frac{\partial}{\partial t}\bar{U}\right)_V = -\oint\!\!\oint_{\partial V}\bar{\bar{F}}\bullet\bar{n}\,\mathrm{d}s + \oint\!\!\oint_{\partial V}\bar{\bar{F}}\bullet\bar{n}\,\mathrm{d}s \tag{3}$$

The inviscid and viscous integration implied by Eq. (3) proceeds on a dual mesh formed by connecting the geometric centers of the physical mesh cells which surround each node. Figure 1 shows construction of this dual auxiliary mesh in two and three dimensions. The state vector of conserved quantities is stored at nodal locations. Expression (3) may be further specialized for application to a specific polyhedral control volume V_i, with planar faces, and may then be approximated by:

$$\frac{\partial\bar{U}_{V_i}}{\partial t} = -\frac{1}{V_i}\sum_{k\in\partial V_i}\bar{\bar{F}}_{I_k}\bullet\bar{S}_k + \frac{1}{V_i}\sum_{k\in\partial V_i}\bar{\bar{F}}_{V_k}\bullet\bar{S}_k$$

$$\frac{\partial\bar{U}_{V_i}}{\partial t} = -\frac{1}{V_i}\sum_{k\in\partial V_i}\bar{Q}_k + \frac{1}{V_i}\sum_{k\in\partial V_i}\bar{\bar{F}}_{V_k}\bullet\bar{S}_k \tag{4}$$

where k denotes the kth face of volume V_i and $\bar{S}_k = [S_x, S_y, S_z]^T_k$ is the surface vector of face k. The first term on the right-hand side of Eq. (4) approximates the inviscid surface integral, whereas the second term balances the viscous fluxes through the faces of the cell around i.

Evaluating the summation for the inviscid flux balance requires computation of the numerical flux $\bar{Q}_k$ through each face k. Using the notation found in Fig. 2, an upwind formulation of this flux may be expressed as

$$\bar{Q}_k = {}^1\!/_2(\bar{\bar{F}}_L + \bar{\bar{F}}_R)\bullet\bar{S}_k + {}^1\!/_2\mathcal{R}_k\bar{\Phi}_k \tag{5}$$

Here $(\ )_L$ and $(\ )_R$ denote conditions to the left and right of face k. Equation (5) separates $\bar{Q}_k$ into a summation of a symmetric term constructed like a central difference operator and a term which adapts this stencil in accordance with local wave propagation.

Two formulations of the flux function $\bar{\Phi}_k$ were suggested by Yee which lead to second-order TVD schemes.[1] The present method is the less dissipative of the two, and relies on the "modified flux" approach of Harten. With $l = 1, \ldots, 5$, the lth component of $\bar{\Phi}_k$ may be written as

$$\Phi^l_k = {}^1\!/_2\Psi(\lambda^l_k)(g^l_L + g^l_R) - \Psi(\lambda^l_k + \gamma^l_k)\,\alpha^l_k \tag{6}$$

with

$$\bar{\alpha}_k = \mathcal{R}^{-1}_k(\bar{U}_R - \bar{U}_L) \tag{7}$$

leading to Harten and Yee's upwind TVD scheme.[10] In Eq. (7) $\bar{\alpha}_k$ would be the difference of the characteristic variables from $(\ )_R$ to $(\ )_L$ if $\mathcal{R}^{-1}_k$ were evaluated at the right and left states individually. Instead, the nonlinear nature of the governing equations forces us to evaluate the inverse of the right eigenvector matrix at face k at an intermediate state, and we employ the approximate Riemann solver developed by Roe to define this state. Here, λ^l are the eigenvalues of the transformed flux Jacobians. If the wave speed λ^l were to vanish the asymmetric term on the right-hand side of Eq. (5) would also vanish and thus violate the entropy condition.[2] To prevent this, $|\lambda|$ in Eq.(6) is thresholded near zero by introducing the functions $\Psi(z)$ and γ^l_k

$$\gamma^l_k \equiv \psi(\lambda^l_k)\begin{Bmatrix}(g^l_R - g^l_L)/\alpha^l_k & \alpha^l_k \neq 0\\ 0 & \alpha^l_k = 0\end{Bmatrix} \tag{8}$$

and

$$\Psi(z) \equiv \begin{Bmatrix}|z| & |z| \geq \delta\\ (z^2+\delta^2)/2\delta & |z| < \delta\end{Bmatrix},\quad \text{with } \delta \ll 1 \tag{9}$$

The limiter function g^l appearing in Eqs. (6) and (9) is presented in Ref. 4. Notice that setting this limiter identically to zero degenerates the entire method to Roe's first-order scheme. The imple-

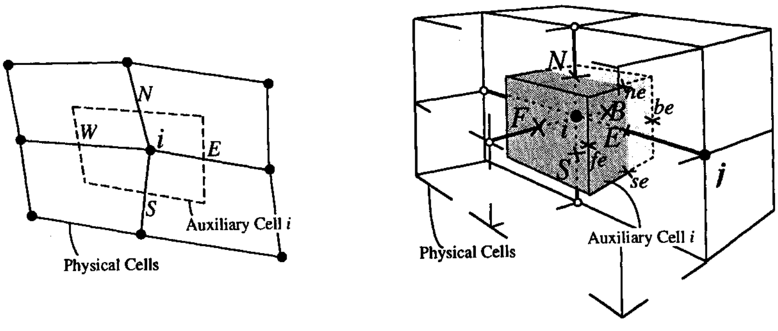

Fig. 1 Formation of dual mesh of auxiliary cells in two and three dimensions.

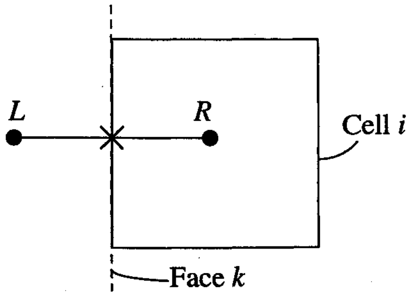

Fig. 2 Cell labels required for construction of the numerical flux $\bar{Q}_k$, through face k of cell i.

mentation of Ref. 17 provides a basis for the current node-based formulation (see also Ref. 6).

Adopting this node-based implementation results in a convenient approach for the construction of an unstructured adaptive algorithm, while still permitting accurate and straightforward treatment of boundaries. In developing this algorithm, the inviscid flux balance of Eq. (3) is evaluated using midpoint quadrature, and thus the numerical flux $\bar{Q}_k$ must be evaluated at the center of each auxiliary cell face.

Figure 3 shows the stencil required by Eqs. (3) and (6) to form the complete numerical flux through the west face of V_i. These expressions emphasize that the first-order ($\bar{g} \equiv 0$) upwind approximation of the inviscid flux requires only nearest neighbor information. However, as shown in the exploded view of this stencil, the complete flux function on the west face of this cell $\bar{\Phi}_W$ (required for a second-order approximation) depends on information from beyond the nearest neighbor. This observation re-emphasizes the fundamental difficulty in the application of higher order upwind methods to unstructured meshes.

This dilemma is alleviated through re-examination of the formulation presented by Eqs. (6–9). These expressions divide the formation of the complete stencil into a sequence of operations within which no single step requires communication beyond the nearest neighbor. By first calculating $\bar{\alpha}_k$ across all of the faces of all cells on the auxiliary mesh and then storing these values in each cell, the limiters $\bar{g}_k$, and therefore the flux function $\bar{\Phi}_k$, may be evaluated by simply interrogating only the auxiliary cell immediately to the west of cell V_i and V_i itself. Breaking up the second-order discrete operator into these two successive steps forms the complete update to the state vector at i with only nearest neighbor inquiries.

This multiple-pass evaluation of the upwind difference stencil is analogous to the splitting of the fourth-order dissipation operator in an unstructured central difference scheme. In such schemes, the fourth difference stencil is too large to be computed with only nearest neighbor information, and so it is usually evaluated by taking the second difference of a set of second differences which were precomputed in a previous sweep.[10] Similarly, the current procedure forms the limiter $\bar{g}$ as a function of the difference of characteristic variables that were precalculated in a prior step. The additional memory requirements implied by this strategy are not severe as will be shown in a subsequent section.

C. Spatial Integration of Viscous Fluxes

The inviscid integration scheme may be extended to the full Navier-Stokes equations by a separate discretization of the viscous components of the flux density tensor. As suggested by the form of Eq. (4), the complete update to any node $\Delta\bar{U}_i$ is a summation of inviscid and viscous contributions.

$$\Delta\bar{U}_i = (\Delta\bar{U}_i)_I + (\Delta\bar{U}_i)_V \tag{10}$$

The viscous discretization uses central difference modeling and the selection of control volumes is similar to that found in Ref. 8. The choice of central differencing allows the scheme to be written extremely compactly, and eventually it involves only node-to-cell and cell-to-node communication in addition to organizing all gather-scatter operations so that they may be easily grouped for rapid processing. These communication issues remain important because the current implementation uses a cell-based data structure (like that in Refs. 7–9) and each physical cell can address only the nodes at its vertices directly. Thus an implementation based on cell-to-node operations is especially convenient.

Its worth noting that this basic discretization is also amenable to edge-based data structures. In fact, Ref. 18 describes the formulation of Hessian and Laplacian operators using only edge-based formulas. For the hexahedral based meshes discussed here, however, this strategy holds little obvious advantage, and so the cell-based formulas were used.

Reference 19 contains a complete development of the viscous discretization in two and three dimensions. The intent of this section is to provide an outline of that discretization so that the important implementational issues may be clearly understood. The complete viscous discretization may be illustrated by considering the formation of the model second derivative term u_{xx} at node i. Recalling the preceding assumption that the control volume V_i has planar faces and applying the Gauss divergence theorem on the surface ∂V_i of auxiliary cell V_i in Fig. 1 gives

$$\begin{aligned}(u_{xx})_i &= \frac{1}{V_i}\oint_{\partial V_i}(u_x)\,n_x\,\mathrm{d}S\\ &= \frac{1}{V_i}\Big[(u_x)_E S_x^E - (u_x)_W S_x^W + (u_x)_N S_x^N - (u_x)_S S_x^S\\ &\quad + (u_x)_F S_x^F - (u_x)_B S_x^B\Big]\end{aligned} \tag{11}$$

The x components of both the unit normal vector n_x and the surface vector S_x are taken oriented in positive coordinate directions. With the auxiliary cells constructed as described earlier, each edge incident on node i will always pierce the face of an auxiliary cell. Thus, Eq. (11) contains one contribution from each edge terminating at i. This observation becomes increasingly important when integrating more general polyhedral control volumes.

Equation (11) makes use of first derivatives u_x at the N, S, E, W, F, and B faces. A second surface integral, similar to this expression, provides each of these quantities at the required midface location. This is accomplished by constructing a secondary set of control volumes which surround each of the edges incident on node i. Figure 4 illustrates the secondary cell constructed around the edge piercing the E face of auxiliary cell in two and three dimensions.

$$\begin{aligned}(u_x)_E &= \frac{1}{V_E}\oint_{\partial V_E} u n_x\,\mathrm{d}S\\ &= \frac{1}{V_i}[ujS_x^j - u_i S_x^i + u_{ne}S_x^{ne} - u_{se}S_x^{se} + u_{fe}S_x^{fe} - u_{be}S_x^{be}]\end{aligned} \tag{12}$$

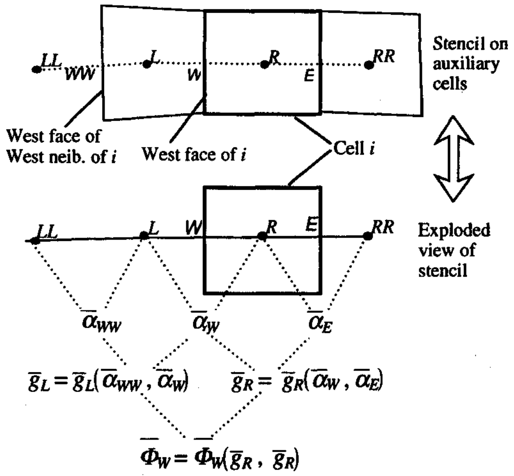
LL
WW
L
W
R
E
RR
Stencil on auxiliary cells
West face of West neib. of i
West face of i
Cell i
LL
L
W
R
E
RR
Exploded view of stencil
$\bar{\alpha}_{WW}$
$\bar{\alpha}_W$
$\bar{\alpha}_E$
$\bar{g}_L = \bar{g}_L(\bar{\alpha}_{WW}, \bar{\alpha}_W)$
$\bar{g}_R = \bar{g}_R(\bar{\alpha}_W, \bar{\alpha}_E)$
$\bar{\Phi}_W = \bar{\Phi}_W(\bar{g}_R, \bar{g}_R)$

Fig. 3 Difference stencil for second-order upwind representation of the flux function $\bar{\Phi}_W$ at the west face of cell i.

where V_E is the volume of the secondary cell. Equation (12) requires the surface vectors for the east secondary cell, and these are constructed by averaging the surface vectors from the faces of the auxiliary cells surrounding nodes i and j (the nodes which define the end points of the edge which pierces the E face of V_i).

$$(\bar{S}^{ne})_{V_E} = {}^1\!/_2(\bar{S}_i^N + \bar{S}_j^N), \qquad (\bar{S}^{se})_{V_E} = {}^1\!/_2(\bar{S}_i^s + \bar{S}_j^s)$$

$$(\bar{S}^{fe})_{V_E} = {}^1\!/_2(\bar{S}_i^F + \bar{S}_j^F), \qquad (\bar{S}^{be})_{V_E} = {}^1\!/_2(\bar{S}_i^B + \bar{S}_j^B) \quad (13)$$

$$(\bar{S}^{j})_{V_E} = {}^1\!/_2(\bar{S}_j^W + \bar{S}_j^E), \qquad (\bar{S}^{ne})_{V_E} = {}^1\!/_2(\bar{S}_i^N + \bar{S}_j^N)$$

Similar constructions around the remaining edges which meet at i form the remaining first derivatives needed to evaluate Eq. (11). The choice of this set of secondary control volumes, and the use of the divergence theorem, insures that the secondary control volumes are closed, and therefore that the viscous discretization remains freestream preserving on any arbitrary mesh. Furthermore, this discretization is completely conservative, and the formulation recognizes and damps odd-even decoupling on stretched or unstretched meshes.[19]

Unstructured implementation of the discretization outlined by Eqs. (11) and (12) essentially reduces to rearranging these expressions to match the data structure. Within the current cell-based structure, the operations comprising these two surface integrals are broken up into contributions from each physical mesh cell. Operations are then performed within each mesh cell and contributions are distributed to each cell's vertices (nodes). This process forms the complete update at the mesh nodes. In visualizing this, note that the first derivatives required by Eq. (11) are located on the faces of the auxiliary cells. These locations correspond to mid-edges of the physical mesh. Thus, the algorithm sweeps through the physical mesh on a cell-by-cell basis, forming the viscous fluxes along the edge of the physical cells, and then balancing these fluxes on the auxiliary cells to form the nodal update. The actual implementation is discussed fully in Ref. 19 and is based on the two-dimensional work of Ref. 20. Implicit in this approach is the fact that under the assumptions of a Newtonian fluid the viscous flux calculation becomes a linear combination of first derivative quantities.

D. Eigenvalue Scaling for High Aspect Ratio Cells

Near wall boundaries, high aspect ratio cells are commonly used to efficiently resolve the boundary layer. Since the artificial dissipation is scaled isotropically with the spectral radius of the flux Jacobian matrices, the damping in the wall normal direction may become excessive. Reference 21 proposed a three-dimensional extension to the two-dimensional variable scaling proposed by Martinelli[22] which adjusts levels of the blended second- and fourth-order smoothing used in conjunction with the central difference option for the convective flux discretization. When the TVD convective dissipation is chosen, this nonisotropic scaling affects only the waves which are subjected to the entropy cutoff. In the current work, all waves are cut in Euler simulations, whereas in viscous simulations, only the nonlinear $(u \pm a)$ eigenvalues are thresholded by the entropy cutoff. In such cases, this scaling takes a form similar to that found in Ref. 22.

III. Adaptation

The adaptation algorithm increases the resolution of the discrete solution by locally reducing the mesh spacing. The algorithm begins by scanning a preliminary (coarse) solution for regions of interest. It then enhances the mesh in those regions through the addition of new computational nodes. These new points are currently added through cell division[7] and the position of all mesh nodes may then be modified by a redistribution step which provides more smoothly varying grid metrics without altering mesh connectivity.

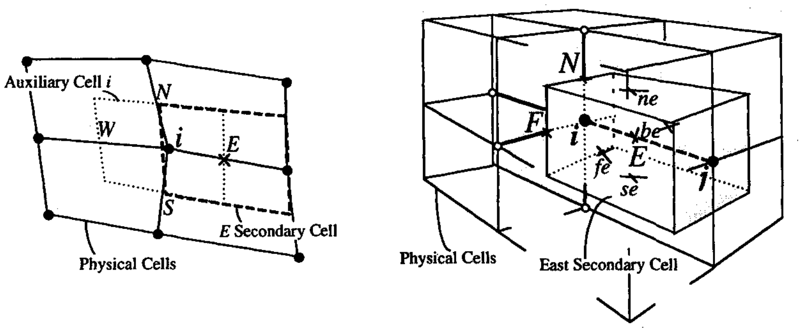

Fig. 4 Construction of secondary cell for the calculation of first derivatives on the E face of the auxiliary cell which surrounds node i (left: two-dimensional, right: three-dimensional).

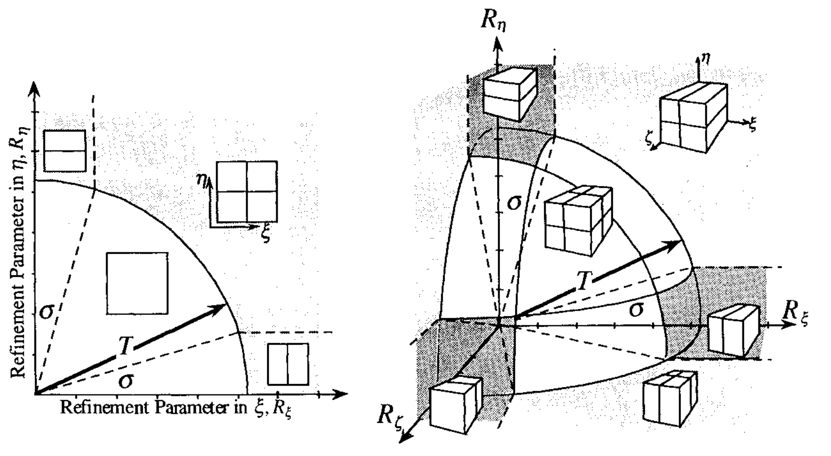

Fig. 5 Schematic of adaptation map for directional division of cells in two and three dimensions.

A. Feature Detection

The crisp shocks afforded by the TVD integration scheme may be detected by undivided differences of several parameters. However, in problems involving high-speed flow where several parameters may jump by orders of magnitude through a shock (e.g., static pressure), there exists a danger of overlooking less prominent features with a crude detection algorithm. Furthermore, recent studies of various feature detection strategies demonstrate that failure to adapt to smooth, less prominent features may lead to an adaptive procedure which actually converges to the wrong solution.[23] For these reasons, a new procedure has been developed which separates shock recognition from the detection of other flow features.

Several other factors contribute to this decision. The current algorithm captures shocks with only two or three cells[6,17] because the upwind difference stencil can change structure to reflect the different nature of the governing equations on either side of such an interface. With a characteristic thickness of only a few mean free paths, a shock is one of the very few flow structures which will always be much smaller than the local mesh dimension (in a continuum situation). Thus, although the adaptation will eventually embed several computational cells across most other flow features, this will not occur at shocks. Such reasoning gives both physical and numerical motivation for treating shock detection in a special manner.

1. *Shock Detection*

A normalized, undivided second difference of pressure provides a parameter which is quite sensitive to both strong and weak shocks. Such a quantity recognizes curvature in the pressure profile across a three-point difference stencil. Thus, it is well suited to recognize the steplike profiles associated with the discrete shocks expected from the TVD scheme.[1,17] Additionally, it performed well in the study of refinement parameters provided in Ref. 7. This second difference is then normalized by a weighted average of local pressures to reduce its magnitude at strong shocks and increase its size at weak ones. The components of the shock refinement parameter R_s are given by

$$R_{s\xi} = \frac{|\delta_\xi^2 p|}{p}, \qquad R_{s\eta} = \frac{|\delta_\eta^2 p|}{p}, \qquad R_{s\zeta} = \frac{|\delta_\zeta^2 p|}{p}$$

Any physical cell with R_s exceeding some threshold (usually 0.05) is tagged for division as a shock cell. Although somewhat empirical in basis, this procedure very reliably locates shocks in trans-, super-, and hypersonic flow. Moreover, the overall detection algorithm is quite insensitive to the precise threshold value chosen. The primary drawback of this parameter lies in the fact that it may mistakenly tag a strong viscous or inviscid feature as a "shock." Nevertheless, most features which display such strong nonlinearity usually warrant detection, and this is not a serious shortfall.

2. *Smooth Feature Detection*

With the cells near strong nonlinearities successfully identified, only the remaining cells are then rescanned for smoother, less prominent features. With these extrema removed, the remaining field may be considered "smooth," and features may be located on a statistical basis as in Ref. 20. Undivided differences of total velocity (scaled as in Ref. 23) and density provide a basis for locating inviscid features. A second sweep using an undivided second difference of velocity magnitude is used to locate regions of rapidly varying shear stress to identify poorly resolved viscous layers in Navier-Stokes simulations. All physical cells containing values of the refinement parameter above a threshold set 20% above the mean are tagged for division.

3. *Directional Division*

The current adaptive procedure identifies not only a feature's location, but also its orientation (as in Ref. 8). Many flow features are predominantly one dimensional and are frequently almost aligned with the coarse base grid. With this information, the hexahedral cells may be divided directionally to avoid excessive resolution in directions where resolution requirements are more relaxed. In three dimensions, this means that a cell may be divided in only one or two directions. Such refinements create only one or three additional cells, rather than the seven which are created if each cell always divides in all three directions.

As the detector scans the field, it stores refinement parameters for each physical cell in all three computational directions. Figure 5 contains a sketch of an adaptation map (in both two and three dimensions) within which all of the cells in a two- or three-dimensional domain will lie. Using the notation on the figure, each cell will have some $[R_\xi, R_\eta, R_\zeta]$ coordinate and will plot on this map. The threshold T mentioned earlier appears as a quarter sphere on this figure, and all cells outside of it become marked for refinement. Furthermore, the location of the cells on this map indicate the directions in which the cells require refinement. The rays which delineate different regions emanate from the origin and are currently set at an angle of 15 deg.

To avoid the formation of excessive interface regions between cells at different levels of refinement, directional adaptation was used only at the finest level of cell division.[6,20] This final division typically creates from 50% (two dimensional) to 75% (three dimensional) of the final nodes in the mesh, and directional division typically reduces the total number of new nodes created by about 50% in both two and three dimensions.

B. Interfaces

An essential complication of hexahedral cell division is the appearance of interfaces between different levels of physical cells. Figure 6 shows a region of divided cells bordering on an undivided region in two and three dimensions. For simplicity, the cells have been divided in all directions. Shown at the left of this figure are the physical cells, and on the right are the auxiliary cells formed by connecting the centers of the physical cells. In two dimensions, this construction results in the formation of triangular auxiliary cells around hanging nodes such as B. In three dimensions the midface nodes (such as B) are surrounded by a pyramidal auxiliary cell and midedge nodes, such as C, are surrounded by prismatic cells. All of these auxiliary cells may be adopted into the framework by permitting a degeneracy in the north surface vector ($\bar{S}_N \equiv 0$) of auxiliary cells B and C and integrating the resulting cells without special treatment.

Since the node-based formulation proceeds by interacting the auxiliary cells, without special care, all cells whose difference stencil crosses the interface will suffer a first-order stretching error in the calculation of derivatives across this interface. Focusing on

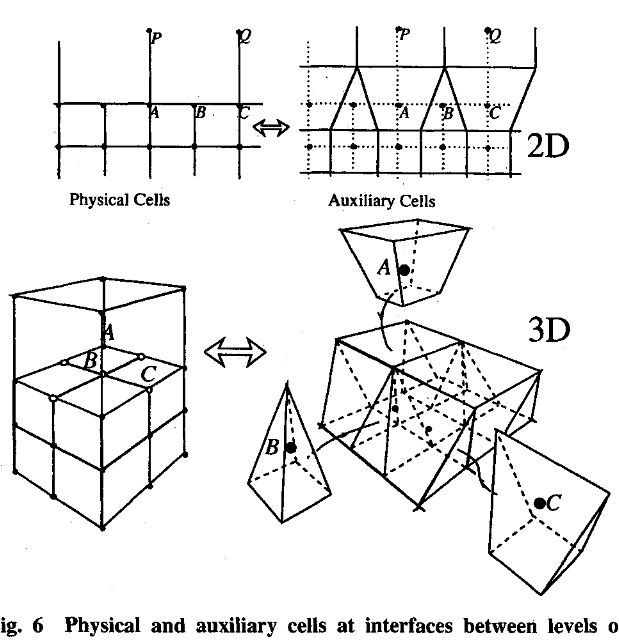

Fig. 6 Physical and auxiliary cells at interfaces between levels of divided cells in two and three dimensions.

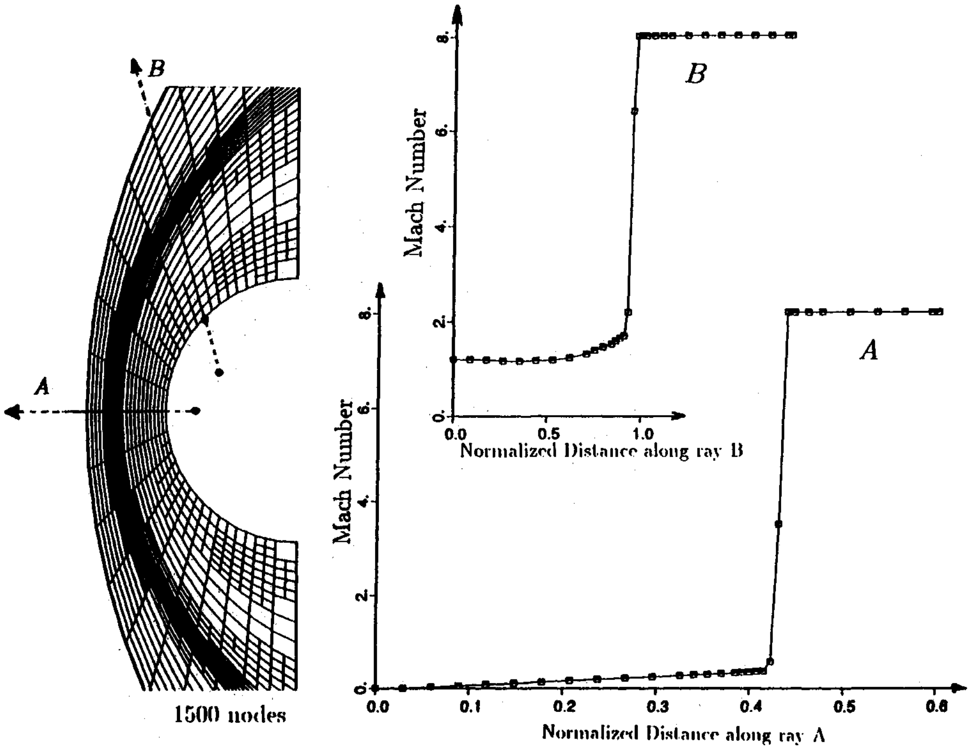

Fig. 7 Final adapted mesh (1500 nodes) and Mach number distributions along rays *A* and *B* for inviscid two-dimensional flow over circular cylinder.

the two-dimensional example in the upper half of the figure, this means that although cells such as *A* and *P* have completely different stencils, in regions where flow properties vary faster than linearly, these differences will be contaminated by mesh stretching. Triangular, prismatic, or pyramidal cells also suffer a stretching error, but, more importantly, their difference stencil is left incomplete. For example, cell *B* has no northern neighbor which gives rise to a degeneracy in the northern extension of *B*'s difference stencil. This deficiency is overcome by linearly interpolating for the missing $\bar{\alpha}$ [Eq.(7)] from the $\bar{\alpha}_N$ on the north faces of *A* and *C* (in the two-dimensional case). The treatment of *B* as a degenerate cell whose northern surface vector is zero retains storage space for the interpolated $(\bar{\alpha}_N)_B$. Such a quantity is required by the limiter on the south face of cell *B* when second-order accuracy is sought.

These ideas form a simple and effective strategy for treating the interfaces which appear in adaptive hexahedral meshes. First, the centers of all physical cells are used to define the auxiliary mesh. Then, if a degenerate cell is encountered while sweeping to calculate the difference of characteristic variables $\bar{\alpha}_k$, the missing $\bar{\alpha}_k$ stem from a combination of the $\bar{\alpha}_k$ on similar faces of its neighboring cells. This treatment permits a complete operator to be formed on the face opposite the degeneracy, and after filling in the missing $\bar{\alpha}_k$, the flux balance for the prismatic or pyramidal cells proceeds without special handling. This simple treatment is both conservative and robust. It suffers only from the same stretching error experienced by any node whose stencil is similarly stretched. The viscous update to the interface nodes proceeds on the same set of control volumes.

IV. Numerical Investigations and Results

The presentation of numerical results is organized to first establish the validity of the solving procedure and then to examine global aspects of the adaptive methodology. The investigations focus on topologically simple example problems which are designed to first establish the accuracy of the inviscid and viscous discretization, and then to examine the ability of the adaptation to correctly locate and resolve structures in complex three-dimensional flows with interacting viscous and inviscid features. Unless specifically noted, all numerical results presented use the Upwind TVD numerical flux.

A. Fundamental Issues

This section begins with a brief examination of shock and boundary-layer resolution, preservation of monotonicity, and other basic issues of the numerical modeling. The first numerical example considers inviscid flow over a two-dimensional circular cylinder exposed to a Mach 8.03 crossflow. The left half of Fig. 7 contains the final adapted mesh (1500 nodes), and line plots of local Mach number are included to the right. These Mach number distributions follow a ray tracing outward from the centerline and along another line inclined 36 deg from vertical. As is evident from the grid, directional division was employed at the finest level of refinement.

All cases presented in this paper were computed using the limiter function g^l taken from Ref. 11 [Eq. (3.51f)] which behaves smoothly and is not overly compressive.

$$g_W^l = \frac{\alpha_L^l \alpha_R^l + \left|\alpha_L^l \alpha_R^l\right|}{\alpha_L^l + \alpha_R^l + \delta} \tag{14}$$

This example demonstrates that the method has successfully retained the favorable shock capturing properties associated with the upwind TVD scheme on structured meshes (see, for example, Refs.1, 4, 11, and 17). Even off centerline—where the shock is not grid aligned—the discrete shock is contained within two cells. Furthermore, the Mach number distributions in each plot cross four grid interfaces, and the conservative treatment described in the preceding section successfully maintains a smooth, nonoscillatory demeanor through these interfaces. Measuring to the sonic point, the shock stand-off distance in this example agrees with the published value to within one grid point.[24]

A second inviscid example examines the full three-dimensional formulation and highlights the utility of directional adaptation in three dimensions. Figure 8 provides an overview of a case in which a double wedge corner flow is simulated at $M_\infty = 2.98$ with

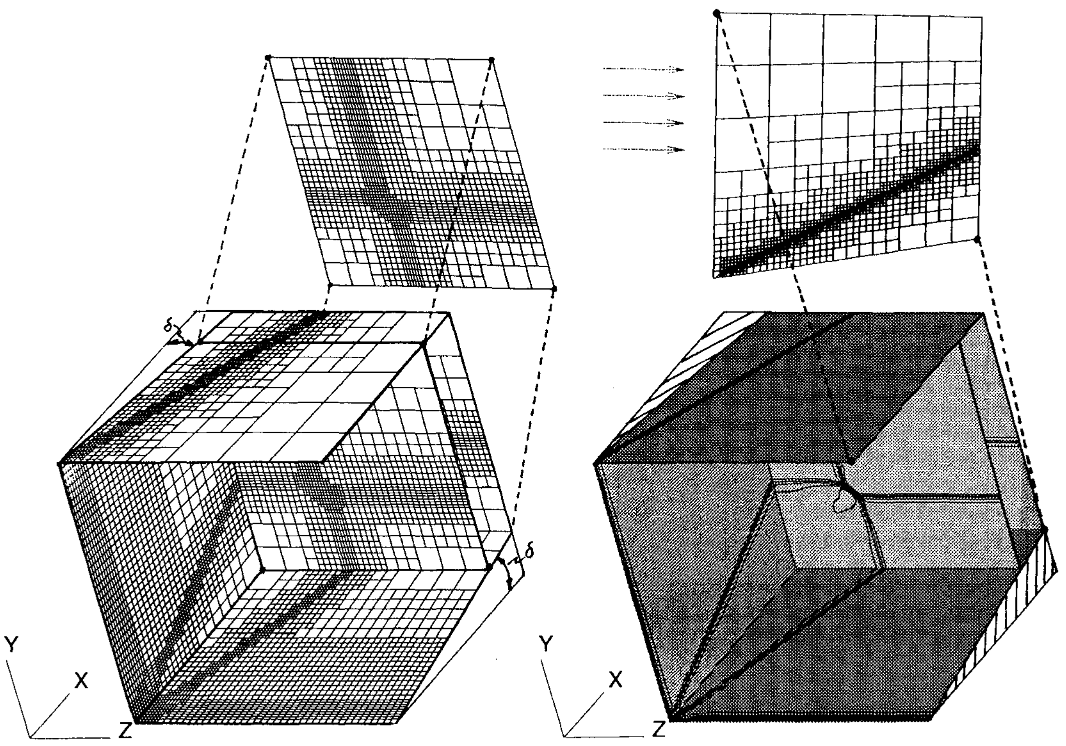

Fig. 8 Inviscid double wedge corner flow test case; $M_\infty = 2.98$, $\delta_1 = \delta_2 = 9.49$ deg, and 70,000 nodes.

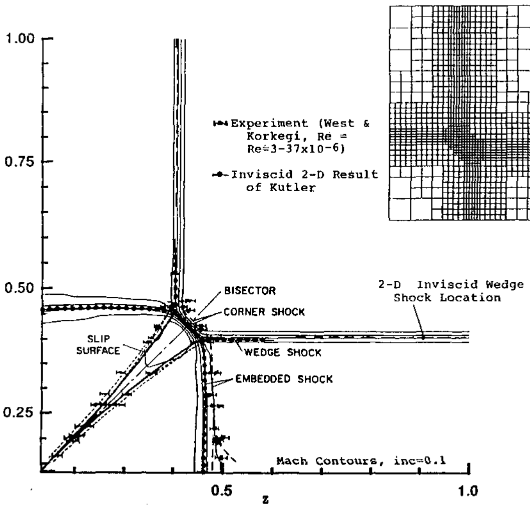

Fig. 9 Mach contours in plane at x = 0.8 superimposed upon shock positions from inviscid gasdynamic theory,[24] experimental data,[25] and previous inviscid calculation.[26]

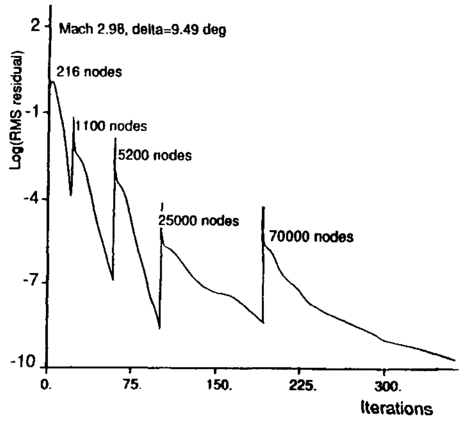

Fig. 10 Convergence history of rms sum of all state vector residuals.

wedge angles $\delta_1 = \delta_2 = 9.49$ deg. The final mesh contains ~70,000 nodes which is 14 times fewer than the roughly 980,000 which would be required to provide the same shock resolution on a globally refined mesh. Directional division was again permitted at the finest mesh level, and the mesh plane depicted above the grid (at x = 0.8) shows that the cells have adapted themselves to match the orientation of the wedge and corner shocks. This plane appears again in Fig. 9, where contours of constant Mach number are superimposed directly on both experimental data[25] and a previous inviscid numerical simulation.[26] Additionally, the undisturbed wedge flow away from the corner interaction allows direct comparison with inviscid gasdynamic theory. The agreement with experiment, theory, and prior calculations results is very good, and the shocks spans no more than two cells. Figure 10 shows a convergence history for this example showing a net reduction in the global residual of approximately nine orders of magnitude.

A subsonic flat plate boundary layer provides an initial assessment of the behavior of the viscous discretization. This quasi-two-dimensional flow was computed at two levels of resolution using both TVD and central difference operators for $M_\infty = 0.5$ and Re_L = 5000. This simulation used an entropy cutoff in the TVD scheme of 0.1 and a fourth difference dissipation coefficient of 1/128 in the central difference calculation. The domain extended 2L upstream, along, and above the plate. Figure 11 displays u-velocity profiles of both schemes with 5 and 13 points in the boundary layer. With only 6 points in the boundary layer, the upwind TVD method has already essentially reproduced the Blasius profile. This result contrasts sharply with that from the central difference scheme at this level of resolution. With 13 points in the boundary layer, the two integrations produce essentially identical results. At this higher level of resolution, the central difference result produces almost no evidence of the "viscous overshoot" evident in the profile with 6 points.

Figure 12 shows skin friction development along the flat plate. This plot reports results for both the central and TVD discretizations at two levels of resolution and compares these with the Blasius relation $C_f' = 0.664(Re_x)^{-1/2}$. With 13 points in the boundary layer (Re_L of 5000), the theoretical skin friction is predicted well by about $Re_x = 100$.

B. Three-Dimensional Viscous Flow

With the basic properties of the algorithm established, Fig. 13 opens the presentation of a more complex example aimed at facilitating an evaluation of the feature detection algorithm. Success or failure of the adaptive strategy rests on the ability of this routine to locate and resolve important structures and smooth regions in the flow. Thus, it is important to evaluate the feature detection algorithm on realistically complex, viscous flows at high resolution.

The test case beginning in Fig. 13 considers Mach 1.2 flow over the 65-deg cropped delta wing tested in Ref. 27 at 20-deg incidence angle. At these conditions, this flow is characterized by a strong steady primary vortex which separates at the sharp leading edge. Underneath this vortex, a crossflow shock develops which induces separation of the boundary layer and "locates" the secondary separation. The tertiary separation lies between the crossflow shock and sharp leading edge. On the windward side, a bow shock develops to turn the flow around the wing. This combination of interacting viscous and inviscid features makes this a discriminating problem for evaluating the capabilities of the feature detection algorithm for realistic, complex three-dimensional flows.

The laminar, half-span calculation used a Reynolds number of 480,000 based on root chord, whereas the experiment was conducted at $Re_C = 2.4$–5.3×10^6. The wing in the calculation was not truncated at the trailing edge, but instead was extended downstream to the outflow boundary (at approximately $x = 1.1c$).

Figure 13 contains an overall view of the computation. The wing is depicted with the mesh on the starboard and density contours on the port side of the delta wing. These contours show the footprints of the primary vortex, crossflow shock, and secondary and tertiary vortices on the wing surface. The final adapted mesh used 999,000 nodes, placing about 260 points chordwise and 150

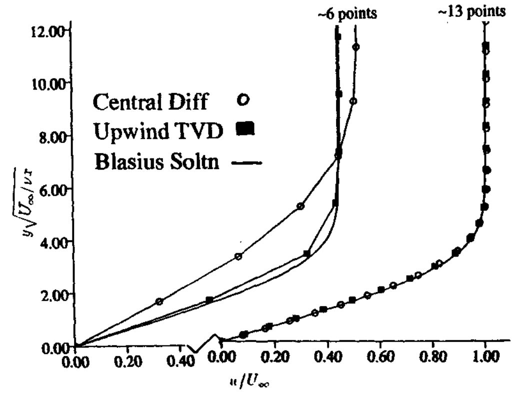

Fig. 11. Flat plate boundary-layer calculation, $M_\infty = 0.5$, $Re_L = 5000$; comparison of u-velocity profile with Blasius solution at various levels of mesh resolution.

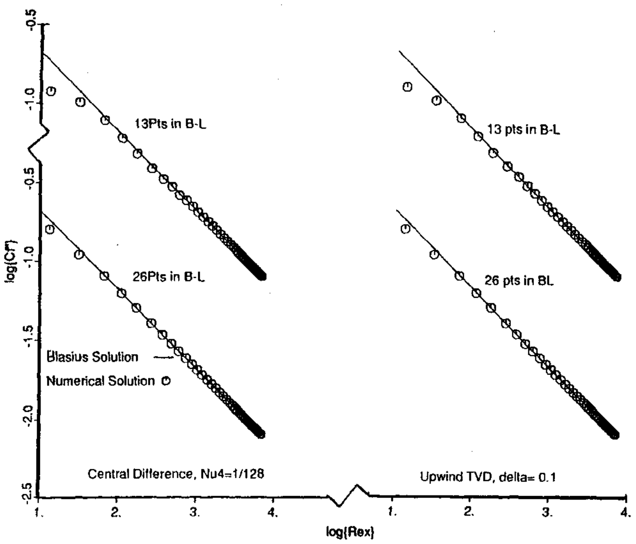

Fig. 12. Flat plate boundary layer calculation, $M_\infty = 0.5$, $Re_L = 5000$; comparison of skin friction with Blasius solution at various levels of mesh resolution.

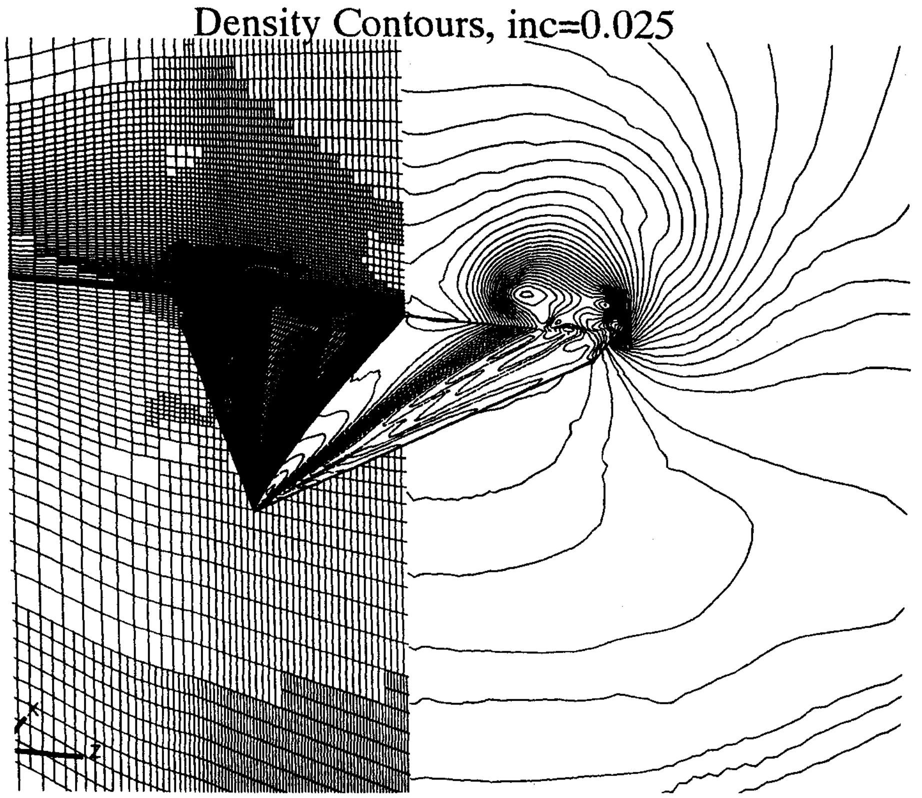

Fig. 13 Navier-Stokes solution of supersonic flow over delta wing of Ref. 27 at $M_\infty = 1.2$, $Re_C = 480,000$, $\alpha = 20$ deg with 999,000 nodes.

points spanwise (at the trailing edge) on the upper surface of the wing. Figure 14 shows further details of the flow as it passes through the plane located at the trailing edge using contours of total pressure loss. The figure clearly shows the adaptation pattern responding to capture the shock triggered separation underneath the primary vortex, as well as resolution of feeding sheet and the entrained tip vortex. This indicates that the feature detector successfully located these weaker features, while still responding to the strong vortical structures and bow shock. Figure 15 contains symmetry plane Mach contours and a view of the corresponding adapted grid whereas Fig. 16 compares this calculation with wind-tunnel results from Ref. 27. The computed surface streamlines reproduce the overall surface shear pattern from the experimental oil flow visualization including formation of the secondary and tertiary separations. The tertiary vortex in the calculation does not appear to form quite as early or as fully as in the experiment, and this is believed to be a Reynolds number effect. Figure 16 also contains a direct comparison of the C_p distribution at $x = 0.8c$ with experimental measurements. Starting at the symmetry plane, the line plot agrees with the data underneath the primary, through crossflow shock and secondary separation point, and begins to differ only in the location of the tertiary separation. Despite the Reynolds number difference, this comparison is quite reasonable. Nevertheless, it should be re-emphasized that this example was completed to focus on the performance of the detection algorithm and was not conducted as a rigorous attempt to reproduce wind-tunnel data.

C. Processing Efficiency and Storage

The three-dimensional viscous solver has been ported to Cray X-MP, Y-MP, and Cray 2 platforms. The main loop is completely vectorized, including both the numerical flux calculation and the computation of viscous fluxes. A simple coloring scheme is implied by sequential sweeps over corresponding faces of all cells. Some of the loops within the viscous flux calculation required "unwinding" by hand to vectorize these routines, but no other special coding was required.

After loading a restart file and preprocessing all geometric calculations, timing tests were run on 500 iterations of the 999,000 node 65-deg cropped delta wing presented in the preceding set of figures. This example contained 50,000 boundary nodes and the boundary conditions were not fully vectorized. With the TVD option chosen, the Navier-Stokes simulation proceeded at 70 μs/node/iteration on single processor of the X-MP (five-stage scheme). Selecting central differencing for the inviscid fluxes results in Navier-Stokes simulations at 30 μs/node/iteration. Careful timings were not completed on either the Cray 2 or the Y-MP.

The solver currently stores 153 words/node for full upwind Navier-Stokes simulations. Of this storage 30 words/node are required for the pre-calculation of the difference of characteristic variables across all cell faces (as discussed in Sec. II), and thus, the central difference scheme requires approximately 125 words/node. On a per-node basis, the scheme requires 3–4 times more storage than an efficiently written structured solver. Adaptive methods hope to offset this by using fewer nodes to obtain discrete solutions of comparable accuracy. The three-dimensional examples presented earlier in this section require roughly an order of magnitude fewer grid points than equivalently resolved structured mesh solutions. Additionally, the use of fewer nodes not only reduces storage requirements, but also decreases the time required per time step, since fewer nodes must be integrated. With the exception of local time stepping, no acceleration techniques were applied to the basic solver. A host of such techniques exist for accelerating Runge-Kutta schemes, and the next section discusses likely candidates for incorporation.

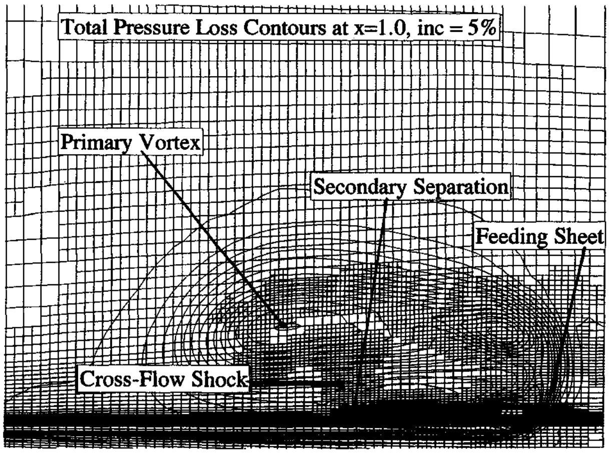

Fig. 14 Total pressure loss contours at trailing edge (x = 1.0) showing shock-induced separation.

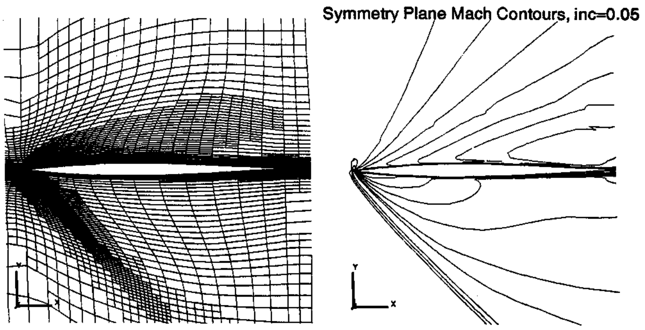

Fig. 15 Mesh and Mach contours in symmetry plane.

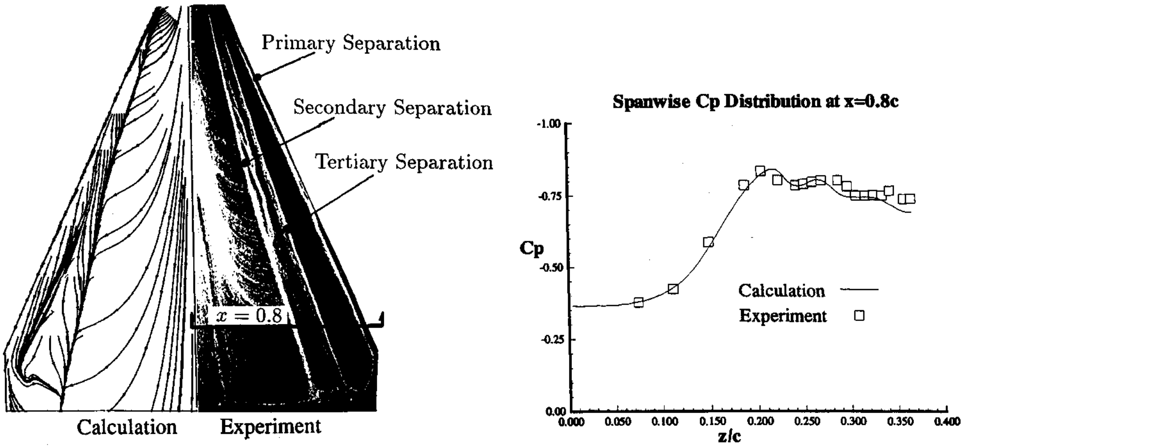

Fig. 16 Right: Results from experimental oil flow visualization[27] (Re_C = 2.4–5.3 × 10^6); left: computed surface streamlines from numerical solution (Re_C = 4.8 × 10^5); and comparison of Cp at x = 0.8 with experimental results from Ref. 27.

V. Summary and Future Work

A second-order upwind method for solution of the three-dimensional Navier-Stokes equations was formulated on adaptively embedded meshes. This implementation permits local adaptation through cell division in response to emerging flow features. The feature detection algorithm separates the detection of shocks from that of other flow phenomena and can divide cells directionally in response to local resolution requirements.

Application of the method to several inviscid and viscous test problems pointed out the ability of the method to provide accurate solution to examples with high resolution requirements. These examples demonstrated that the unstructured method retains the crisp, nonoscillatory shock representation associated with TVD schemes on structured meshes. Simulation of laminar, viscous flows highlighted the ability of the full Navier-Stokes solver to properly model viscous flow by comparison with both theoretical and experimental results in both two and three dimensions.

With the preliminary developmental work on this solver complete, further efforts will focus on improving the flexibility and utility of the method. The Runge-Kutta time stepping and embedded mesh topology make multigrid acceleration a natural choice for increasing the convergence efficiency of the method. Such work should also examine possibilities for domain decomposition and implicit temporal integration. Finally, future investigations will also explore the application of the method to complex configurations using block-structured initial meshes.

Acknowledgments

Financial support for this work, which was accomplished within a scientific exchange program between the U.S. Air Force and the Deutsche Forschungsanstalt für Luft und Raumfahrt e.V. during the author's tenure as a Visiting Scientist at DLR Braunschweig, was provided by the SEEP program and is gratefully acknowledged. Computing support was provided by both the DLR and the Phillips Laboratory Supercomputing Center at Kirtland AFB. The author would like to sincerely thank N. Kroll for his invaluable contributions to this work and to the exchange as a whole.

References

[1] Yee, H. C., "Upwind and Symmetric Shock-Capturing Schemes," NASA-TM 86842, May 1987.

[2] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 52, No. 2, 1983, pp. 357–393.

[3] Roe, P. L., "Approximate Reimann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–393.

[4] Kroll, N., Gaitonde, D., and Aftosmis, M., "A Systematic Comparative Study of Several High Resolution Schemes for Complex Problems in High Speed Flows," AIAA Paper 91-0636, Jan. 1991.

[5] Ramanurti, R., and Löhner, R., "Simulation of Subsonic Viscous Flows Using Unstructured Grids and a Finite Element Solver," AIAA Paper 90-0702, Jan. 1990.

[6] Aftosmis, M., and Kroll, N., "A Quadrilateral Based Second-Order TVD Method for Unstructured Adaptive Meshes," AIAA Paper 91-0124, Jan. 1991.

[7] Dannenhoffer, J. F., III, and Baron, J. R., "Adaptation Procedures for Steady State Solution of Hyperbolic Equations," AIAA Paper 84-0005, Jan. 1984.

[8] Kallinderis, Y., "A Finite Volume Navier-Stokes Algorithm for Adaptive Grids," *International Journal for Numerical Methods in Fluids,* Vol. 15, No. 2, 1992, pp.193–217.

[9] Kallinderis, Y., and Baron, J. R., "Adaptation Methods for a New Navier-Stokes Algorithm," *AIAA Journal,* Vol. 27, No. 1, 1989, pp. 37–43.

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

[11] Yee, H. C., and Harten, A., "Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates," *AIAA Journal*, Vol. 25, No. 2, 1987, pp. 266–274.

[12] Roe, P. L., and Beard, L., "An Improved Wave Model for Multidimensional Upwinding of the Euler Equations," *Proceedings of the 13th International Conference on Numerical Methods in Fluid Dynamics* (Rome), Springer-Verlag, Heidelberg, Germany, 1992, pp. 135–139.

[13] Rumsey, C., van Leer, B., and Roe, P. L., "A Grid-Independent Approximate Reimann Solver with Applications to the Euler and Navier-Stokes Equations," AIAA Paper 91-0239, Jan. 1991.

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

[15] Durlofsky, L. J., Osher, S., and Enquist, B., "Triangle Based Adaptive Stencils for the Solution of Hyperbolic Conservation Laws," *Journal of Computational Physics,* Vol. 98, No. 1, 1992, pp. 64–73.

[16] Jameson, A., "A Vertex Based Multigrid Algorithm for 3-Dimensional Compressible Flow Calculations," *Numerical Methods for Compressible Flows—Finite Difference, Element, and Volume Techniques,* edited by T. E. Tezduyar and T. Hughes, Applied Mechanics Div. 78, American Society of Mechanical Engineers, New York, 1986.

[17] Kroll, N., and Rossow, C., "A High Resolution Cell Vertex TVD Scheme for the Solution of the Two and Three Dimensional Euler Equations," *Proceedings of the 12th Conference on Numerical Methods in Fluid Dynamics* (Oxford, UK), Springer-Verlag, Heidelberg, Germany, 1990, pp. 442–446.

[18] Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations," AGARD Rept. 787, May 1992.

[19] Aftosmis, M., "Viscous Flow Simulation Using an Upwind Method for Hexahedral Based Adaptive Meshes," AIAA Paper 93-0772, Jan. 1993.

[20] Kallinderis, Y. G., "Adaptation Methods for Viscous Flows," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, 1988.

[21] Radespiel, R., Rossow, C., and Swanson, R. C., "Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," *AIAA Journal,* Vol. 28, No. 8, 1990, pp. 1464–1472.

[22] Martinelli, L., "Calculations of Viscous Flows with a Multigrid Method," Ph.D. Thesis, Dept. of Mechanical and Aerospace Engineering, Princeton Univ., Princeton, NJ, 1987.

[23] Warren, G. P., Anderson, W. K., and Krist, S., "Grid Convergence for Adaptive Methods," AIAA Paper 91-1592, June 1991.

[24] Liepmann, H. W., and Roshko, A., *Elements of Gas Dynamics,* Wiley, New York, 1957, pp. 104–105.

[25] West, J. E., and Korkegi, R. H., "Supersonic Interactions in the Corner of Intersecting Wedges at High Reynolds Numbers," *AIAA Journal,* Vol. 12, No. 5, 1972.

[26] Kutler, P., "Supersonic Flow in the Corner by Two Intersecting Wedges," *AIAA Journal,* Vol. 12, 1974, pp. 577, 578.

[27] Erickson, G. E., Schreiner, J. A., and Rogers, L. W., "On the Structure, Interaction, and Breakdown Characteristics of Slender Wing Vortices at Subsonic, Transonic, and Supersonic Speeds," AIAA Paper 89-3345, June 1989.