A 3D Finite-Volume Scheme for the Euler Equations on Adaptive Tetrahedral Grids

VIJAYAN P. AND Y. KALLINDERIS

Department of Aerospace Engineering, WRW 201 C, University of Texas at Austin, Austin, Texas 78712

Received November 13, 1992; revised July 21, 1993

The paper describes the development and application of a new Euler solver for adaptive tetrahedral grids. Spatial discretization uses a finite-volume, node-based scheme that is of central-differencing type. A second-order Taylor series expansion is employed to march the solution in time according to the Lax-Wendroff approach. Special upwind-like smoothing operators for unstructured grids are developed for shock-capturing, as well as for suppression of solution oscillations. The scheme is formulated so that all operations are edge-based, which reduces the computational effort significantly. An adaptive grid algorithm is employed in order to resolve local flow features. This is achieved by dividing the tetrahedral cells locally, guided by a flow feature detection algorithm. Application cases include transonic flow around the ONERA M6 wing and transonic flow past a transport aircraft configuration. Comparisons with experimental data evaluate accuracy of the developed adaptive solver. © 1994 Academic Press, Inc.

1. INTRODUCTION

Considerable progress has been made over the past years on development and application of computational fluid dynamics methods for fluid flow simulations. However, computation of flows around three-dimensional bodies remains a major issue in CFD. Generation of a body-conforming grid is a difficult task [1].

In the case of *structured grids*, spatial orientation of the elements allows an (i, j, k) index to be employed for denoting the grid points. It becomes increasingly difficult to model the computational grid as the flow geometries encountered become more complex such as in 3D flow simulations about an entire aircraft configuration. The multiblock strategy, introduced by [2], has been used for such complex geometries. This concept is to break the flow field into several smaller blocks and then generate separate meshes in each individual block. Grid generation around configurations like a complete aircraft entails the need for a large number of such blocks. The problem of defining the blocks and their interfaces becomes increasingly difficult as the number of blocks increases.

In the case of *unstructured grids*, one can dispense with the need for the grid points to be in any order. The tetrahedron is the simplest element in 3D and therefore it is the most flexible in covering complex topologies [1, 3–5], wherein no single natural coordinate direction exists. A compromise between the two approaches has been the employment of semi-unstructured prismatic elements [6]. Such grids are employed within the viscous regions of a flow domain in order to capture the strong directionality of the viscous stresses, while tetrahedral grids are applied within the inviscid regions.

The spatial derivative approximations used in these algorithms can usually be considered as central space differencing with added dissipation or as flux split upwind differencing. Central space differencing schemes require the addition of special dissipation terms in order to capture the shock waves as well as to suppress the solution oscillations. Upwind schemes do not require any explicitly added dissipation, but they are usually more expensive and complex when compared to the central space differencing schemes. The required computation time per grid cell by a numerical scheme is an important issue in cases of 3D tetrahedra grid computations due to the relatively large number of cells that are required. Furthermore, central-differencing schemes can adjust the smoothing operators in order to increase the effectiveness of multigrid accelerators, which is more difficult to do with the upwinding operators. The issue of the type of spatial differencing is not settled within the community and both types of approaches are currently pursued. In the present work a central differencing scheme is presented. However, smoothing operators have a special upwind-like form.

Resolution of the computational mesh plays a crucial role on the accuracy of computations. However, generation of a grid which both fits the flow geometry and resolves the local flow features is quite difficult, and even impossible in some cases. In general the selection of the grid that is to be used in a numerical simulation is determined a priori starting the solution procedure, and quite often the grid is modified by the user, in order to improve the results. Adaptive grid algorithms are flexible enough to adjust the grid during the solution procedure without intervention by the user.

0021-9991/94 $6.00

Frequently, the regions that require high resolution are very small compared to the size of the overall computational domain. Local grid embedding consists of division of the cells in order to reduce the truncation error and to have a more equal distribution of it throughout the solution domain. Quadrilateral meshes have been employed for inviscid flows [7, 8], as well as for turbulent flows [9–11]. Once of the most serious problems with using quadrilateral meshes has been the presence of grid-interfaces, which require special numerical treatment [9, 12]. Such treatments can be quite complicated, and especially so in three dimensions. Triangular meshes have also been employed with grid embedding [13, 14]. An attractive feature of unstructured grids is the ease with which adaptive refinement can be included. The flow solver requires no further modifications when employing an adapted grid. Up to the present, very few such algorithms have been developed [4, 16].

In the present work, a central space differencing, node-centered Euler scheme for tetrahedral grids is presented. It is a one-step Lax–Wendroff-type, conservative scheme. The scheme is formulated so that all operations are edge-based, which reduces the computational work significantly. A significant work that involved central differencing, the Lax–Wendroff-type scheme, was by [17], which uses structured grids with quadrilateral elements, and a similar scheme was also implemented for three-dimensional meshes using hexahedral elements [18]. The scheme was also extended to the 2D Navier–Stokes equations [9, 10]. The adaptive grids are created by division of the tetrahedral cells. Isotropic as well as directional division of the cells provides considerable flexibility in increasing resolution of the grid with the minimum possible number of cells. Application cases include transonic flow around the ONERA M6 wing and transonic flow around an aircraft configuration. Comparisons with experimental data evaluate accuracy of the developed solver.

In the following sections, the spatial as well as the temporal discretizations are described, followed by a presentation of the smoothing operators. Effectiveness of the adaptive algorithm is demonstrated via applications for which experimental data exist.

2. GOVERNING EQUATIONS

The governing equations to be solved is the system of time-dependent Euler equations for a perfect gas which combines the equations of mass, momentum, and energy,

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}+\frac{\partial \mathbf{H}}{\partial z}=0. \tag{1}$$

The state vector **U** and the flux vectors **F**, **G**, **H** are expressed in terms of the conservation variables, namely, density ρ, x, y, z-momentum and energy as

$$\mathbf{U}^{\mathrm{T}}=(\rho \quad \rho u \quad \rho v \quad \rho w \quad E) \tag{2}$$

$$\mathbf{F}=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ Eu+pu\end{pmatrix},\qquad \mathbf{G}=\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ Ev+pv\end{pmatrix},$$
$$\mathbf{H}=\begin{pmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ Ew+pw\end{pmatrix}, \tag{3}$$

where E is the total energy, related to the other variables by an equation of state which, for a perfect gas, is

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2+w^2).$$

3. TEMPORAL DISCRETIZATION

The solution at any particular node, say 0, at time level $n+1$ can be expressed in terms of the solution at time level n using a Taylor series expansion,

$$\mathbf{U}_0^{n+1}=\mathbf{U}_0^n+\delta\mathbf{U}_0^n$$
$$\delta\mathbf{U}_0^n=\mathbf{U}_0^{n+1}-\mathbf{U}_0^n=\left(\frac{\partial \mathbf{U}}{\partial t}\right)_0^n \Delta t$$
$$+\left(\frac{\partial^2 \mathbf{U}}{\partial t^2}\right)_0^n \frac{\Delta t^2}{2}+O(\Delta t^3) \tag{4}$$

The temporal derivatives in the above expression are evaluated in terms of the spatial derivatives using the governing equations according to the Lax–Wendroff approach. The finite-volume method integrates the Euler equation (1) on the control volume Ω_0 enclosing a particular node, say 0, which is enveloped by the boundary $\partial\Omega_0$.

$$\int_{\Omega_0}\left(\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}+\frac{\partial \mathbf{H}}{\partial z}\right)d\Omega=0 \tag{5}$$

which is rewritten using the divergence theorem as

$$\int_{\Omega_0}\left(\frac{\partial \mathbf{U}}{\partial t}\right)d\Omega=-\int_{\partial\Omega_0}(\mathbf{F}n_x+\mathbf{G}n_y+\mathbf{H}n_z)\,dS \tag{6}$$

$$\left(\frac{\partial \mathbf{U}}{\partial t}\right)_0 \Omega_0=-\int_{\partial\Omega_0}(\mathbf{F}n_x+\mathbf{G}n_y+\mathbf{H}n_z)\,dS \tag{7}$$

$$\left(\frac{\partial \mathbf{U}}{\partial t}\right)_0=-\frac{1}{\Omega_0}\int_{\partial\Omega_0}(\mathbf{F}n_x+\mathbf{G}n_y+\mathbf{H}n_z)\,dS, \tag{8}$$

where n_x, n_y, n_z are the components of the unit vector normal to the area elements dS of the boundary surface $\partial\Omega_0$.

The second-order temporal derivative at the node 0 is evaluated as

$$\int_{\partial\Omega_0}\left(\frac{\partial^2\mathbf{U}}{\partial t^2}\right)d\Omega=-\int_{\Omega_0}\frac{\partial}{\partial t}\left(\frac{\partial\mathbf{F}}{\partial x}+\frac{\partial\mathbf{G}}{\partial y}+\frac{\partial\mathbf{H}}{\partial z}\right)d\Omega \quad (9)$$

which, after changing the order of differentiation, is recast using the divergence theorem as

$$\int_{\Omega_0}\left(\frac{\partial^2\mathbf{U}}{\partial t^2}\right)d\Omega=-\int_{\partial\Omega_0}\left(\frac{\partial\mathbf{F}}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial t}\right)n_x+\left(\frac{\partial\mathbf{G}}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial t}\right)n_y+\left(\frac{\partial\mathbf{H}}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial t}\right)n_z\,dS \quad (10)$$

$$\left(\frac{\partial^2\mathbf{U}}{\partial t^2}\right)_0=-\frac{1}{\Omega_0}\int_{\partial\Omega_0}(\tilde{\mathbf{A}}n_x+\tilde{\mathbf{B}}n_y+\tilde{\mathbf{C}}n_z)\frac{\partial\mathbf{U}}{\partial t}\,dS, \quad (11)$$

where

$$\tilde{\mathbf{A}}=\left(\frac{\partial\mathbf{F}}{\partial\mathbf{U}}\right),\qquad \tilde{\mathbf{B}}=\left(\frac{\partial\mathbf{G}}{\partial\mathbf{U}}\right),\qquad \tilde{\mathbf{C}}=\left(\frac{\partial\mathbf{H}}{\partial\mathbf{U}}\right) \quad (12)$$

are the Jacobian matrices.

4. SPATIAL DISCRETIZATION

The surface integrals that appear in Eqs. (8) and (11) are to be evaluated on the boundary of the control volume of each node in the grid. There are different approaches in defining the control volumes around the nodes.

4.1. *Dual Mesh*

The spatial terms that appear in Eqs. (8) and (11) are evaluated by employing special volumes that form the *dual* mesh. The dual mesh is formed by constructing non-overlapping volumes, referred to as dual cells, around each node. The dual cells represent the control volume associated with the respective node. The dual mesh, for a two-dimensional unstructured grid, is shown with dashed lines in Fig. 1. The mesh is constructed by connecting the mid-points of the edges and the centroids of the triangular elements that constitute the grid and henceforth dividing each triangle into three quadrilaterals of equal areas. The finite-volume around any node, say 0, is constituted by the union of all the quadrilaterals which share that node.

Analogously, the dual mesh for a tetrahedral grid is constructed by dividing each tetrahedron into four hexahedra of equal volumes, by connecting the mid-edge points, face-centroids, and the centroid of the tetrahedron. Figure 2 shows a tetrahedron 0–1–2–3 with the two hexahedral cells 0–E1–F2–C–F3–E4–F1–E5 and 3–E2–F2–C–F4–E6–F1–E5 that constitute a portion of the dual cells around the nodes 0 and 3, respectively. The control volume around a node 0 is thus constituted by a polyhedral hull which is the union of all such hexahedra that share that node. The quadrilateral faces that constitute the dual mesh may not all be planar.

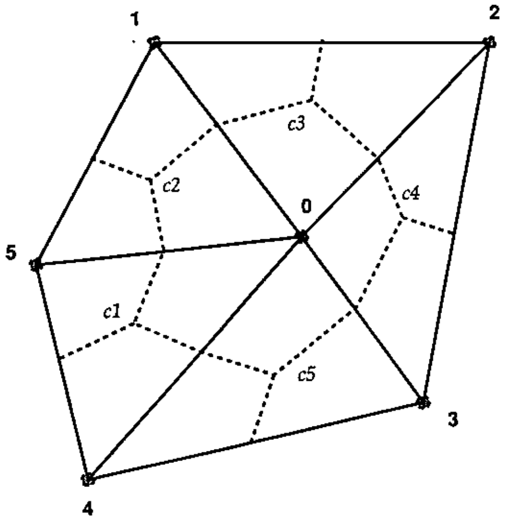


FIG. 1. Dual mesh, shown in dashed lines, for a triangular grid.

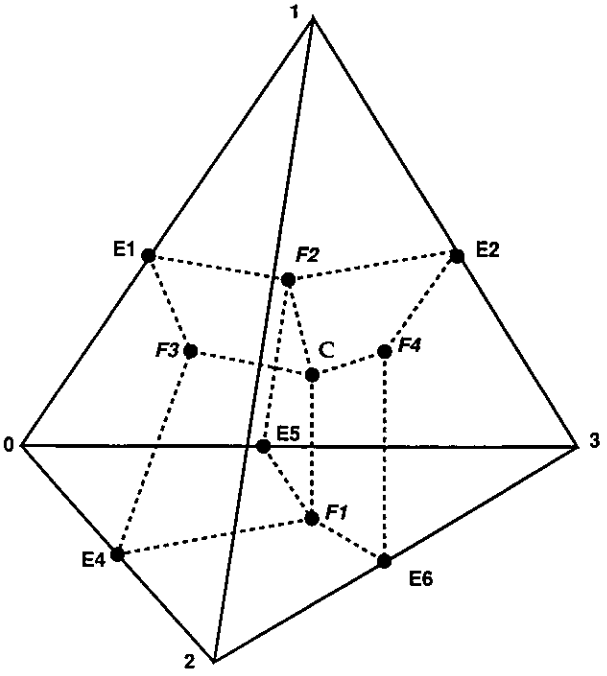


FIG. 2. Dual mesh, shown in dashed lines, for a tetrahedral grid.

4.2. *Flux Evaluation Using Edge-Based Operations*

The surface integral term in Eq. (8) represents the mass, momentum, and energy flux across the faces of the control volume around the node 0. The equation is written in the discrete form as

$$-\frac{1}{\Omega_0}\int_{\partial\Omega_0} (\mathbf{F}n_x + \mathbf{G}n_y + \mathbf{H}n_z)\, dS = -\frac{1}{\Omega_0}\sum_k (\mathbf{F}S_x + \mathbf{G}S_y + \mathbf{H}S_z)_k, \tag{13}$$

where the summation is over all the dual mesh faces that constitute the boundary of the control volume around the node 0. The areas S_x, S_y, S_z are projections of the dual face.

The flux evaluation can be cast into edge-based operations. Consider an edge **i**, constituted by the nodes, 0, $N(i)$. The quadrilateral faces of the dual mesh that are connected to the edge at its mid-point P are shown in Fig. 3. The number of such quadrilateral faces connected to an edge depends on the number of cell neighbors for that edge. As an illustrative case, the edge in Fig. 3 is shown to have four quadrilateral faces. The projections of the area $\bar{\mathbf{A}}_\mathbf{i}$ associated with the edge **i** are evaluated in terms of those of the quadrilateral face areas, $\bar{\mathbf{a}}_1$, $\bar{\mathbf{a}}_2$, $\bar{\mathbf{a}}_3$, $\bar{\mathbf{a}}_4$, as

$$(A_i)_x = \sum_{j=1}^{4} (a_j)_x, \quad (A_i)_y = \sum_{j=1}^{4} (a_j)_y, \quad (A_i)_z = \sum_{j=1}^{4} (a_j)_z. \tag{14}$$

The projections are computed such that the area vector always points outward from the control volume surface associated with any node. The boundary of the control-volume around the node 0 is constituted by the union of such areas $\bar{\mathbf{A}}_\mathbf{i}$ associated with each edge **i** that share the node 0. Thus, the summation over the dual mesh faces in Eq. (13) is equivalent to a summation over the edges of the grid and the fluxes are evaluated on the dual mesh faces associated with each edge. This eliminates a significant amount of computational work as the number of edges is much less than the number of faces in an unstructured grid.

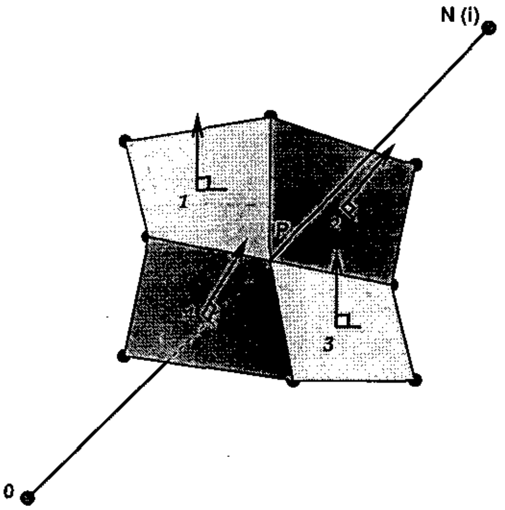


FIG. 3. Dual mesh faces attached to an edge.

In order to evaluate the contribution of each edge to the flux across the control-volume faces of a node 0, the flux vectors **F**, **G**, **H** on the mid-point P of each edge **i** are obtained by taking the average of the flux vectors evaluated at the nodes 0, $N(i)$ on either ends of the edge using the known state variables at these nodes. This strategy has been shown to be equivalent to a finite-element Galerkin approximation which is second-order accurate in space [19]. Thus, the contribution of the edge **i** to the fluxes across the faces of the control volume surrounding the node 0 is given by

$$\mathbf{F}_p(A_i)_x + \mathbf{G}_p(A_i)_y + \mathbf{H}_p(A_i)_z. \tag{15}$$

The fluxes are thus evaluated on an edge-wise basis and conservation is enforced by producing a positive flux contribution to one node and an equally opposite contribution to the other node that constitutes the edge.

Similarly, the surface integral in Eq. (11), which is used to evaluate the second-order temporal derivative by the Lax–Wendroff approach, can be expressed in the discrete form as

$$-\frac{1}{\Omega_0}\int_{\partial\Omega_0} (\tilde{\mathbf{A}}n_x + \tilde{\mathbf{B}}n_y + \tilde{\mathbf{C}}n_z)\frac{\partial\mathbf{U}}{\partial t}\, dS = -\frac{1}{\Omega_0}\sum_{k=1}^{n} (\tilde{\mathbf{A}}S_x + \tilde{\mathbf{B}}S_y + \tilde{\mathbf{C}}S_z)_k \left(\frac{\partial\mathbf{U}}{\partial t}\right)_k, \tag{16}$$

where the summation is over all dual mesh faces that constitute the boundary of the control volume around the node 0. Similar to the evaluation of the first-order temporal derivative, the above operations can be cast as a summation over the edges.

The first-order temporal derivatives evaluated at the nodes 0, $N(i)$ are averaged to obtain the value of $(\partial\mathbf{U}/\partial t)$ at the mid-edge point P. The state vectors known at the nodes are averaged to get the state vector $\mathbf{U}_\mathbf{p}$ at the mid-edge point P with which the Jacobians $\tilde{\mathbf{A}}_i$, $\tilde{\mathbf{B}}_i$, $\tilde{\mathbf{C}}_i$, are computed,

$$A_{lm} = \left(\frac{\partial F_l}{\partial U_m}\right)\bigg|_{\mathbf{U}=\mathbf{U}_p}, \qquad B_{lm} = \left(\frac{\partial G_l}{\partial U_m}\right)\bigg|_{\mathbf{U}=\mathbf{U}_p}, \qquad C_{lm} = \left(\frac{\partial H_l}{\partial U_m}\right)\bigg|_{\mathbf{U}=\mathbf{U}_p}. \tag{17}$$

The contribution of the edge **i** to the evaluation of the second-order temporal derivative at the node 0 is then given by

$$(\tilde{\mathbf{A}}_p(A_i)_x + (\tilde{\mathbf{B}}_p(A_i)_y + (\tilde{\mathbf{C}}_p(A_i)_z)\left(\frac{\partial \mathbf{U}}{\partial t}\right)_p. \tag{18}$$

The contribution, $(\delta \mathbf{U}_0^n)_c$, of the convective flux terms to the total change $\delta \mathbf{U}_0^n$ at node 0 is obtained by substituting Eqs. (15) and (18) into Eq. (4):

$$\begin{aligned}(\delta \mathbf{U}_0^n)_c = &\frac{-1}{\Omega_0}\sum_{i=1}^{n} \Delta t\,(\mathbf{F}_p(A_i)_x + \mathbf{G}_p(A_i)_y + \mathbf{H}_p(A_i)_z)\\ &+\frac{-1}{\Omega_0}\sum_{i=1}^{n}\frac{\Delta t^2}{2}\\ &\times(\tilde{\mathbf{A}}_p(A_i)_x + \tilde{\mathbf{B}}_p(A_i)_y + \tilde{\mathbf{C}}_p(A_i)_z)\left(\frac{\partial \mathbf{U}}{\partial t}\right)_p.\end{aligned} \tag{19}$$

4.3. *Data Structure*

The number of nodes in a typical tetrahedral mesh is approximately five to six times smaller than the number of cells. As a consequence, a vertex-based scheme appears to require less storage compared to a cell-centered scheme. Minimization of the storage requirement is one of the main issues in the development of the present scheme, which stores the state-vector values at grid-nodes. From Eq. (19) it can be seen that all the operations pertaining to the evaluation of the fluxes and the dissipation terms (as will be seen in the next section) can be performed in a single loop through the edges. Hence, an edge-based data structure is a natural choice for the solver. The data structure is constituted of pointers that give the area projections associated with every edge as well as the nodes associated with the edge. As the solver is node-centered, the state vectors are stored at the nodes. These pointers provide all the information that is needed to evaluate the expressions in the relevant equations. The advantages of the edge-based data structures have also been presented by [15, 20].

5. UPWIND-LIKE ARTIFICIAL DISSIPATION

Upwind schemes for solving the hyperbolic equations in the conservation form rely on the theory of characteristics and account for the proper wave propagation directions while differencing the spatial derivatives. These schemes have shown good capability to capture the shocks without oscillations. The upwind connection to artificial dissipation in central differencing schemes is brought forth in [21], using a simple one-dimensional analogy. Adding second-difference dissipation to a second-order accurate central differencing scheme is shown to produce a first-order upwind scheme and adding a fourth-difference dissipation is shown to produce a second-order accurate upwind scheme. The motivation behind the dissipation modelling in the present work is to formulate it in such a manner as to simulate the implicit dissipation terms of the upwinding schemes, without increasing the computation cost of the algorithm.

Considering the one-dimensional Euler equation,

$$\frac{\partial \mathbf{U}}{\partial t} + \tilde{\mathbf{A}}\frac{\partial \mathbf{U}}{\partial x} = 0, \qquad \tilde{\mathbf{A}} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}, \tag{20}$$

given any two states $\mathbf{U}_\mathbf{L}$, $\mathbf{U}_\mathbf{R}$, the flux difference can be uniquely expressed as

$$\mathbf{F}_\mathbf{R} - \mathbf{F}_\mathbf{L} = \sum_k \alpha_k \lambda_k \mathbf{e}_k, \tag{21}$$

where $\mathbf{e}_k$ are the right eigenvectors of $\tilde{\mathbf{A}}$. The term α_k in the summation represents the strength of each wave and λ_k represents the kth eigenvalue of $\tilde{\mathbf{A}}$ (or the speed corresponding to that wave). Using this expression and accounting for the sign of the eigenvalues, the flux vector at any intermediate state **I** between **L** and **R** can be expressed as

$$\mathbf{F}_\mathbf{I} = \tfrac{1}{2}(\mathbf{F}_\mathbf{L} + \mathbf{F}_\mathbf{R}) - \tfrac{1}{2}\sum_k \alpha_k\,|\lambda_k|\,\mathbf{e}_k \tag{22}$$

which can be recast into the form

$$\mathbf{F}_\mathbf{I} = \tfrac{1}{2}(\mathbf{F}_\mathbf{L} + \mathbf{F}_\mathbf{R}) - \tilde{\mathbf{A}}_r(\mathbf{U}_\mathbf{R} - \mathbf{U}_\mathbf{L}), \tag{23}$$

where $\tilde{\mathbf{A}}_r$ is Roe's matrix [22].

The flux vector at the mid-edge point P was taken as the average of the flux vectors at the two nodes of the edge. This is equivalent to evaluating $\mathbf{F}_\mathbf{I}$ using the first term in Eq. (23). Hence, the dissipation terms are modelled so as to be similar to the second term of the equation, as this corresponds to the implicit smoothing term of the upwinding scheme. A simplified form of Eq. (23) is obtained by replacing $\tilde{\mathbf{A}}$ with $\rho(\tilde{\mathbf{A}}) = |u| + c$, the maximum eigenvalue of Roe's matrix. This ensures that the dissipation terms do not dwindle down to zero near the stagnation or the sonic points. The contribution, $(\delta \mathbf{U}_0^n)_{s2}$, of shock smoothing terms to the change $\delta \mathbf{U}_0^n$ at the node 0 is given as

$$\begin{aligned}(\delta \mathbf{U}_0^n)_{s2} = &\frac{\Delta t}{\Omega_0}\sum_{i=1}^{n}(\rho(\tilde{\mathbf{A}}_\mathbf{r})\,S_x + \rho(\tilde{\mathbf{B}}_\mathbf{r})\,S_y + \rho(\tilde{\mathbf{C}}_\mathbf{r})\,S_z)_i\\ &\times(\mathbf{U}_{\mathbf{N(i)}} - \mathbf{U}_\mathbf{0}),\end{aligned} \tag{24}$$

where $\rho(\tilde{\mathbf{A}}_\mathbf{r})$, $\rho(\tilde{\mathbf{B}}_\mathbf{r})$, $\rho(\tilde{\mathbf{C}}_\mathbf{r})$ are the spectral radii of Roe's matrices corresponding to the flux vectors **F**, **G**, **H** evaluated for the edge **i**. The eigenvalues are evaluated at the Roe averaged quantities [23]. The shock smoothing term is evaluated similar to the convective fluxes on an edge-wise basis.

The background smoothing terms are modelled in a similar fashion. Instead of the first difference of state vectors as used in Eq. (24), a difference of the accumulated first difference over the edges sharing a node is taken. This is in concert with the one-dimensional analogy wherein such a difference is equivalent to the fourth-difference operator at the nodes. The contribution, $(\delta \mathbf{U}_0^n)_{s4}$, of background smoothing terms to the change $\delta \mathbf{U}_0^n$ at node 0 is given as

$$(\delta \mathbf{U}_0^n)_{s4} = \frac{-\Delta t}{\Omega_0} \sum_{i=1}^{n} (\rho(\tilde{\mathbf{A}}_r)\, S_x + (\rho(\tilde{\mathbf{B}}_r)\, S_y + (\rho(\tilde{\mathbf{C}}_r)\, S_z)_i \times (\delta \mathbf{U}_{\mathbf{N}(i)} - \delta \mathbf{U}_0), \tag{25}$$

where

$$\delta \mathbf{U}_0 = \sum_k (\mathbf{U}_{\mathbf{N}(\mathbf{k})} - \mathbf{U}_0) \tag{26}$$

is the accumulated first difference over all the edges k sharing the node 0. The background smoothing terms are evaluated on an edge-wise basis as well.

The total change $\delta \mathbf{U}_0^n$ at the node 0 is given by

$$\delta \mathbf{U}_0^n = (\delta \mathbf{U}_0^n)_c + \sigma_2(\Delta P_0)(\delta \mathbf{U}_0^n)_{s2} + \sigma_4(1 - \Delta P_0)(\delta \mathbf{U}_0^n)_{s4}; \tag{27}$$

ΔP is the pressure switch that is used to turn the shock smoothing and the background smoothing on at the appropriate regions. For any node 0, the pressure switch is computed as

$$(\Delta P_0) = \frac{\sum_{i=1}^{n} (P_{N(i)} - P_0)}{\sum_{i=1}^{n} (P_{N(i)} + P_0)}; \tag{28}$$

the summation is over all the edges that share the node 0. The pressure switch is normalized by the maximum value over the domain so that $0 \leqslant \Delta P \leqslant 1$. When evaluated as above, ΔP has a value close to zero in the smooth regions of the flow and it has a value close to unity near the regions of flow discontinuities that are characterized by a pressure jump. The coefficient σ_2 is an empirical parameter that controls the amount of shock smoothing. The shock smoothing is turned on in the regions of flow that have a sharp variation in flow parameters, such as near the shocks, and is turned off elsewhere. The coefficient σ_4 is an empirical parameter that controls background smoothing. The background smoothing is turned on in the smooth regions of the flow and is turned off near the shock regimes using the pressure switch.

6. LOCAL TIME STEPPING

The solution at each node is advanced in time using local time steps. A CFL stability limitation is applied for the convective terms. The viscous-like smoothing term can have appreciable magnitude at shock regions. As a consequence, a stability limitation that combines both the inviscid and the diffusion limitations is applied. The time-step restriction for the 1D wave equation is $\Delta t \leqslant \Delta x/|u| + c$, while the restriction for the 1D diffusion equation is $\Delta t \leqslant \frac{1}{2} \Delta x^2/v$, where in this case $v = \sigma_2 \, \Delta P$. The formula for the time-step, Δt_0, for any node 0 is then given by

$$\Delta t_0 = \omega \frac{V_0}{A_x + A_y + A_z + D}, \tag{29}$$

where

$$A_x = (|u_0| + \alpha_0)\, S_{x0}$$
$$A_y = (|v_0| + \alpha_0)\, S_{y0}$$
$$A_z = (|w_0| + \alpha_0)\, S_{z0}$$

and

$$D = 2\sigma_2 \, \Delta P_0 \frac{V_0}{S_{x0} + S_{y0} + S_{z0}}. \tag{30}$$

In the above expression u_0, v_0, w_0 are the velocity components at the node 0, α_0 is the speed of sound, and ΔP is the pressure switch. The area terms S_{x0}, S_{y0}, S_{z0} are the projected areas of the dual volume around the node 0 in the x, y, and z directions, and they are given by

$$S_{x0} = \tfrac{1}{2} \sum_{j=1}^{n} |S_x|_j, \qquad S_{y0} = \tfrac{1}{2} \sum_{j=1}^{n} |S_y|_j, \qquad S_{z0} = \tfrac{1}{2} \sum_{j=1}^{n} |S_z|_j, \tag{31}$$

where the summation is over all the dual mesh faces that constitute the boundary of the control volume around the node 0. A value of the factor $\omega = 0.5$ has been employed.

7. BOUNDARY CONDITIONS

Three types of conditions have been applied for the cases considered in the present work. Those are (i) flow tangency at wall, (ii) far field, and (ii) symmetry. The flow tangency condition is imposed by extrapolating the velocity at the center of a boundary cell to the corresponding boundary face and then subtracting the component normal to the solid surface. Density and pressure values are extrapolated from the cell-center to the boundary face. Characteristic boundary conditions are applied at the far-field boundaries. The characteristic variables that are employed are the Riemann invariants, the entropy, and the two velocity components that are tangent to the boundary. Lastly, the symmetry condition is applied by simply extrapolating the state-vector from the cell-center to the corresponding boundary face.

8. ADAPTIVE GRID REFINEMENT

A dynamic grid adaptation algorithm has been previously developed for 3D unstructured grids. The algorithm is capable of simultaneously un-refining and refining the appropriate regions of the flow regime by detecting the local flow *features*. In the case of inviscid flows, the dominant flow *features* may be shock waves, expansion waves, or vortices. The regions of existence of such features are not known a priori and they have to be detected. A feature detector senses the flow *features* that are present in different regions and guides the adaptive algorithm to embed these regions if the existing grid spacing in such regions is not sufficient for resolving the local fow variations. The detector visits the regions of the grid that were embedded at earlier passes of grid adaptation and checks if strong flow *features* are still prevailing in these regions. If the features have moved away from these regions, as is common in unsteady flow situations, the detector guids the adaptive algorithm to un-refine the grid locally in those regions of the grid.

8.1. *Feature Detection*

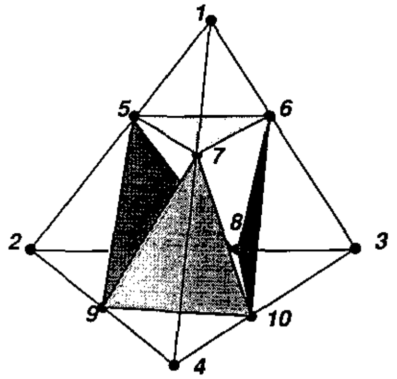
The feature detector uses velocity differences and velocity gradients across the edges as the parameters for sensing the flow *features*. The threshold values for the parameters are set based on the distribution of the parameters which is characterized by the average (S_{ave}) and the standard deviation (S_{sd}) of the respective parameters, where S is any detection parameter [10]. In the present work, velocity differences and velocity gradients are used as the detection parameters. The following relations are used to set the threshold parameters for refinement:

$$S_{\text{ref}\cdot\text{th}} = S_{\text{ave}} + \alpha S_{\text{sd}}.$$

The average and the standard deviation are defined as

$$S_{\text{ave}} = \frac{\sum_{i=1}^{n\ \text{edges}} S_i}{N\,\text{edges}}, \qquad S_{\text{sd}} = \sqrt{\frac{\sum_{i=1}^{n\ \text{edges}} S_i^2}{N\,\text{edges}}}.$$

The value of the parameter α is chosen empirically, a typical value of the parameter being 0.3. The edges that have a detection parameter value greater than the threshold value are flagged to be refined. Following the edge flagging, cells that are having four or more of their edges flagged to be divided are marked for refinement.



CORNER CHILD CELLS (nodes 1,5,6,7 2,5,8,9 3,6,8,10 4,7,9,10)

INTERIOR CHILD CELLS (nodes 5,6,7,8 5,7,8,9, 6,7,8,10 7,8,9,10)

FIG. 4. Octree division of a tetrahedron.

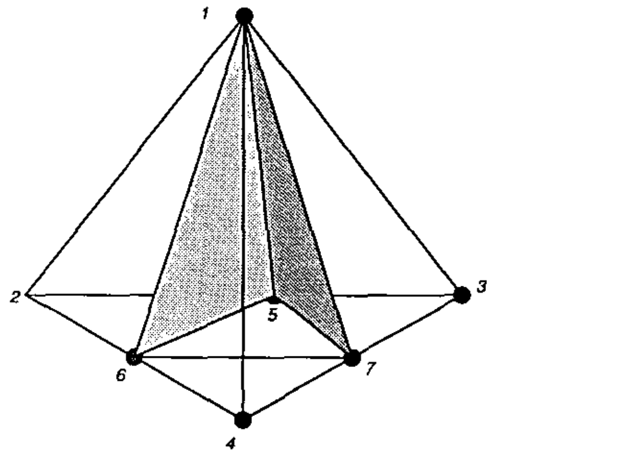
8.2. *Cell Division Strategies*

The different strategies usually employed for embedding a tetrahedron and the methods of cell division employed in the adaptive algorithm used in the present work are discussed in detail in [24].

Octree Division. The tetrahedra that are flagged for refinement are embedded by the octree cell division that divides the *parent* cell into eight *children*, as shown in Fig. 4, by inserting mid-edge nodes on the parent cell edges. The four corner child cells are similar to the parent cell. The four interior child cells are formed by dividing the interior octahedron, constituted by the nodes 5–6–7–8–9–10, by the shortest diagonal.

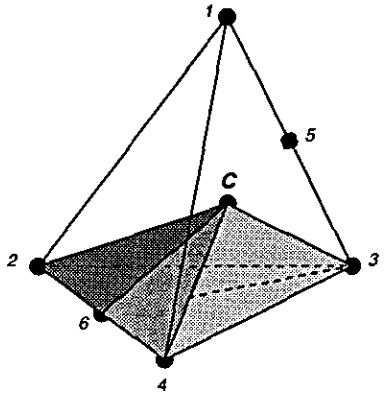
After the octree division of a portion of the grid cells, the resulting grid contains a number of cells that were initially not flagged for division but eventually are left with mid-edge nodes on some or all of their six edges due to refinement in the neighboring cells. These are termed as the *interface cells* as they constitute the border between the divided and the undivided cells and their mid-edge nodes are termed as *hanging nodes*. Numerical schemes usually employ normal tetrahedral cells with four corner nodes and significant changes are necessary in order for the schemes to be applied to such cells with additional *hanging nodes*. This is not desired, as then the adaptive algorithm becomes dependent on the specific numerical scheme that is employed. Hence, a special method of cell division has been incorporated in the adaptive algorithm which eliminated such *interface* cells. There are different configurations in which these hanging nodes appear in the interface cells.

Directional Division. In the case in which all the hanging nodes are appearing on the edges of the same face, the interface cell is directionally divided into four children as shown in Fig. 5. There are four possible cases of such division



FOUR CHILD CELLS (nodes 1,2,5,6 1,3,5,7 1,4,6,7 1,5,6,7)

FIG. 5. Directional cell division into four children when all three hanging nodes are on the same face.

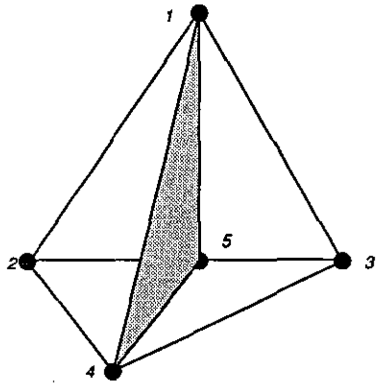


CHILDREN CELLS ON FACE 2,3,4
- nodes C,3,4,6
- nodes C,2,3,6

FIG. 7. Centroidal node dividsion of an interface cell.

depending on which one of the faces the hanging nodes appear. If there is a hanging node appearing on only one of the six edges, the interface cell is henceforth divided into two children as shown in Fig. 6. Depending on the edge that has the hanging node, there are six possible cases of such a division.

Centroidal Node Division. If the *hanging node* configuration is any different from the ones discussed above, the interface cell is treated by introducing a centroidal node and dividing the cell accordingly, as illustrated in Fig. 7. It shows a case of a cell with two hanging nodes, on edges 1–3 and 2–4. A node C is introduced at the centroid of the cell and connected to the corner nodes as well as to the *hanging* nodes, thus forming tetrahedral child cells. This approach is general enough to handle the different hanging node configurations that arise.



TWO CHILD CELLS (nodes 1,2,4,5 1,3,4,5)

FIG. 6. Directional cell division into two children when there is only one hanging node.

The grid adaptation algorithm did not use spline techniques to better obtain the coordinates of the newly inserted nodes on the wall boundary surface of the grid. The coordinates were obtained by linear interpolation only.

9. NUMERICAL RESULTS

Two flow cases are employed in order to provide an assessment of accuracy, robustness, and computer requirements of the developed Euler solver on adaptive tetrahedral grids. The first case is transonic inviscid flow around the ONERA M6 wing. Two different initial grids have been used to obtain the flow solutions on this configuration. The second case considers the low-wing transport (LWT) aircraft. The initial grids used for all the computations have been generated by an advancing front grid generation method [25].

All the computations were performed on a CRAY Y-MP. The code was vectorized to run at a speed of about 100 Mflops. The memory required for the solver was 30 words/node. It should be noted that this memory requirement is quite small for an unstructured grid Euler solver. The values of the smoothing coefficients have been $\sigma_2 = 10^{-2}$ for shock-capturing and $\sigma_4 = 10^{-3}$ for background smoothing.

9.1. *ONERA M6 Wing*

The ONERA M6 wing is considered for evaluating accuracy of adaptive flow solutions. This configuration has been used as a benchmark case for evaluating the accuracy of several Euler methods. The wing has a leading edge sweep of 30°, an aspect ratio of 3.8, a taper ratio of 0.56, and symmetrical airfoil sections. The wing has a root chord of 0.67

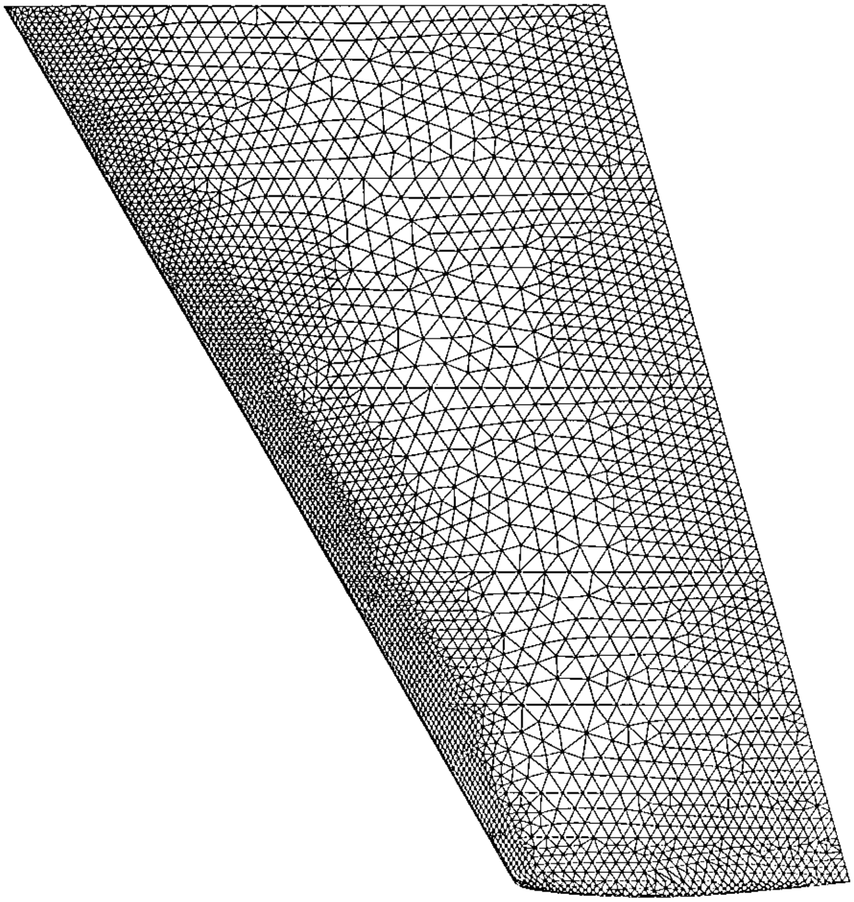


FIG. 8. Triangulation of the ONERA wing upper surface (initial fine grid).

and a semi-span of 1.0 with a rounded tip. The computational domain is bounded by a rectangular box with boundaries at $-6.5 \leqslant x \leqslant 11.0$, $0.0 \leqslant y \leqslant 2.5$, and $-6.5 \leqslant z \leqslant 6.5$. Inviscid, transonic flow solutions were computed at $M_\infty = 0.84$ and angle of attack $\alpha = 3.06°$.

Adaptive Solution with an Initial Fine Grid. The initial mesh employed comprises of 231,507 cells and 42,410 nodes. The triangulation on the wing surface is shown in Fig. 8. The solutions are started from the freestream conditions being specified everywhere. Figure 9 shows the flow solution after 4600 iterations on the initial grid. Mach number contour lines on the upper surface of the wing are shown, plotted using an increment of $\Delta M = 0.02$. The solution clearly features a λ shock that is formed by the two in-board shocks which merge together to form a single strong shock in the tip region of the wing. The fore shock is captured reasonably well, whereas the aft shock appears to be more diffused. This is due to resolution being less in that region of the grid. The convergence history for the solution obtained on the initial grid is shown in Fig. 10.

The initial grid is now adaptively embedded in the regions of the local flow features. Figures 14 and 15 show the triangulation on the wing surface and the symmetry planes with the embedded regions of the grid being denoted with the darker shades. Velocity differences and the velocity gradients were used as the detection parameters. The adapted grid has 833,613 cells and 144,722 nodes. It is seen that grid embedding is aligned along the λ shock. Furthermore, there is more embedding along the aft shock than

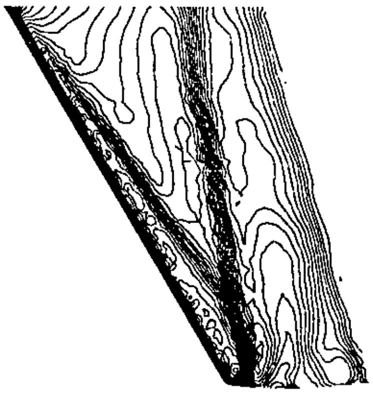


FIG. 9. Mach number contour lines on the wing upper surface. Solution obtained on the initial fine grid after 4600 iterations ($\Delta M = 0.02$).

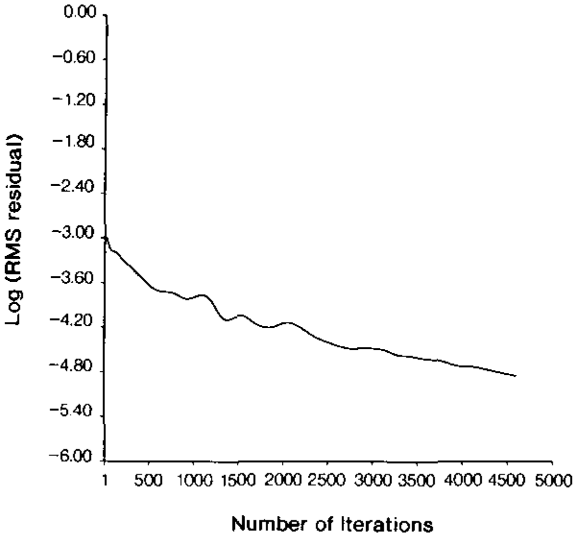


FIG. 10. Convergence history for the solution obtained on the initial fine grid for the ONERA M6 wing.

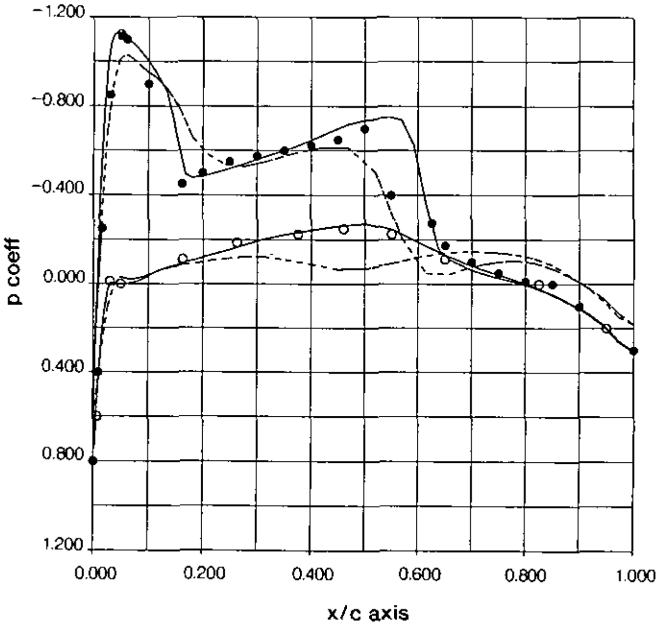


FIG. 12. Isometric view of the triangulation on the wing upper surface and the symmetry plane corresponding to the once adapted fine grid.

along the fore shock as the former is more smeared in the initial solution. There is also an appreciable amount of embedding in the leading edge region of the wing as the flow undergoes rapid acceleration from the stagnation point and reaches the peak Mach number of about 1.50 within 10% chord at all span-wise locations. There is also some embedding on the symmetry plane near the leading edge region and near 75% chord which shows the presence of the weak aft shock at the latter location.

The solution obtained on the initial grid is interpolated to the new grid points and this is used as the starting solution for the adapted grid. Figures 16 and 17 show the solution obtained on the adapted grid. Mach number contour lines on the upper surface of the wing and on the symmetry plane are shown, plotted using an increment of $\Delta M = 0.02$. It is seen from the figures that both the fore and the aft shocks have sharpened to an appreciable extent compared to the corresponding solution plots on the initial grid (Fig. 9). It is also observed that the aft shock on the symmetry plane is much sharper.

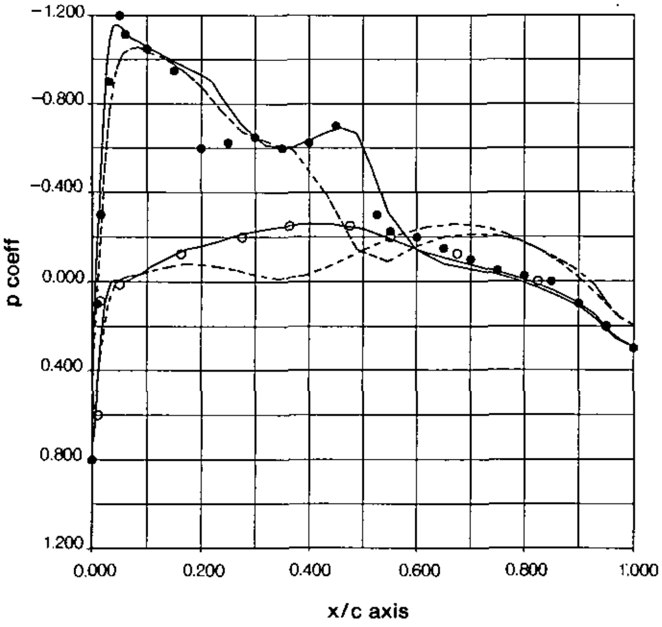


FIG. 11. Triangulation on the wing upper surface corresponding to once adapted grid.

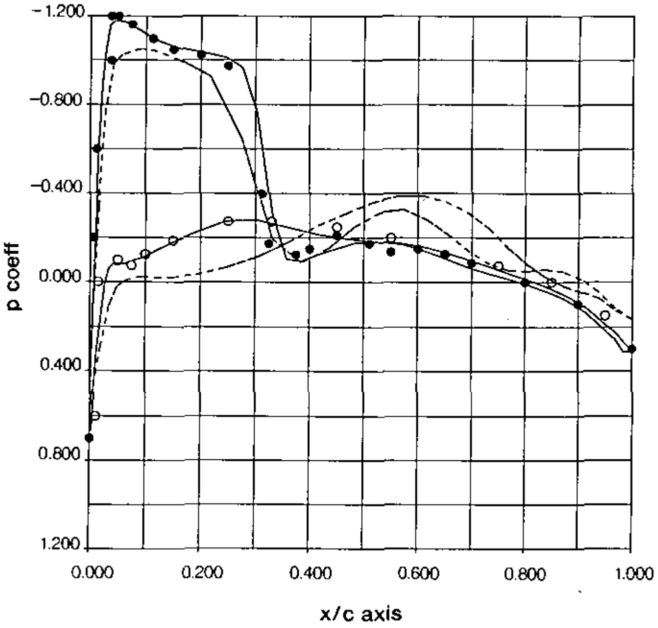
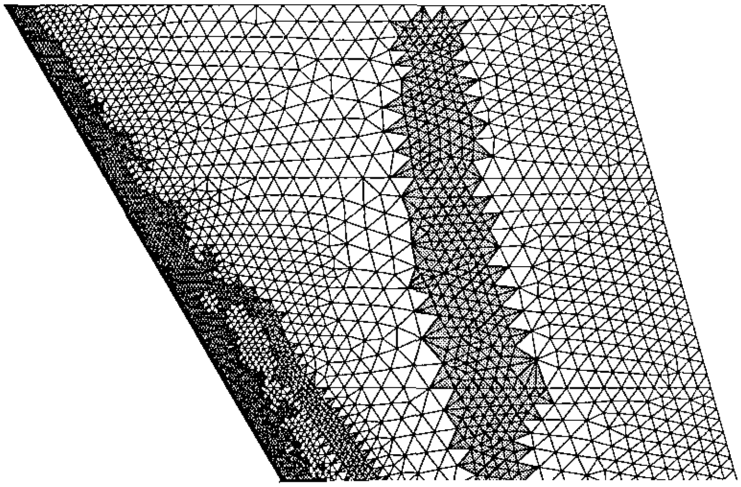


FIG. 13. Mach number contour lines on the wing upper surface. Solution obtained on the once adapted fine grid ($\Delta M = 0.02$).



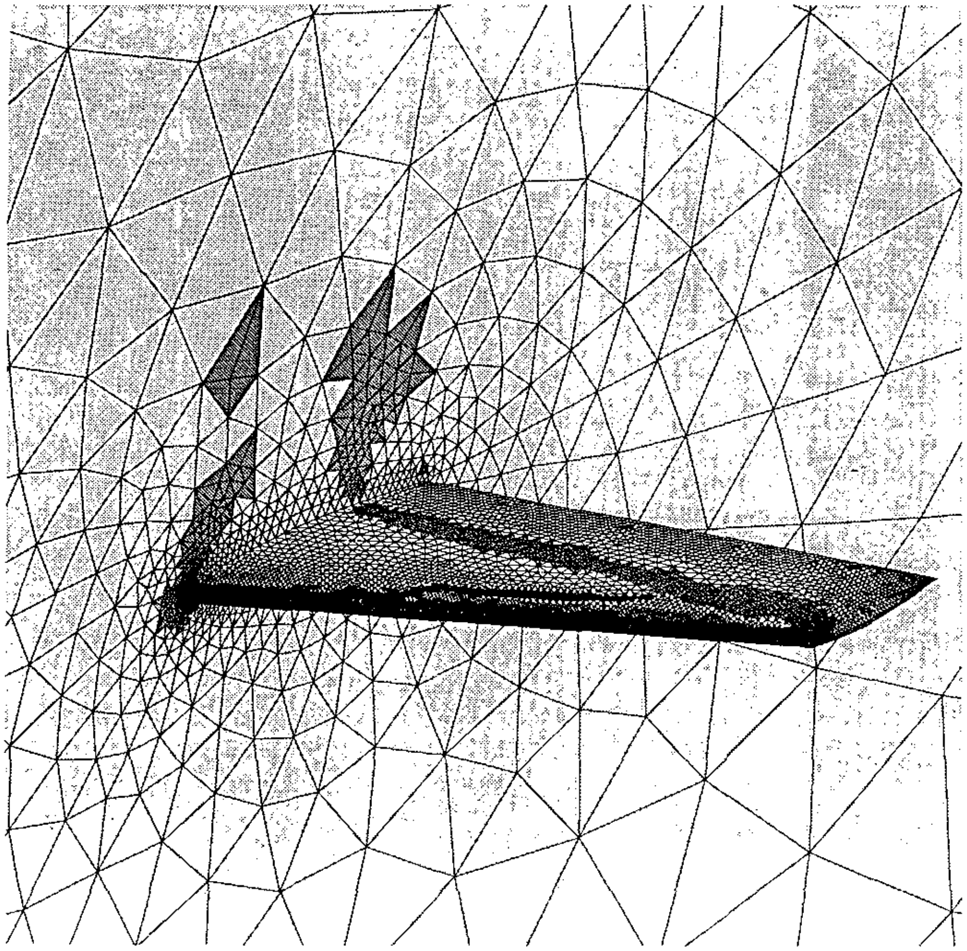


FIG. 15. Pressure coefficients comparison on the wing surface at $\eta = 0.44$ spanwise location: •, Experimental values (upper surface); ◦, experimental values (lower surface); ———, one level adapted grid.

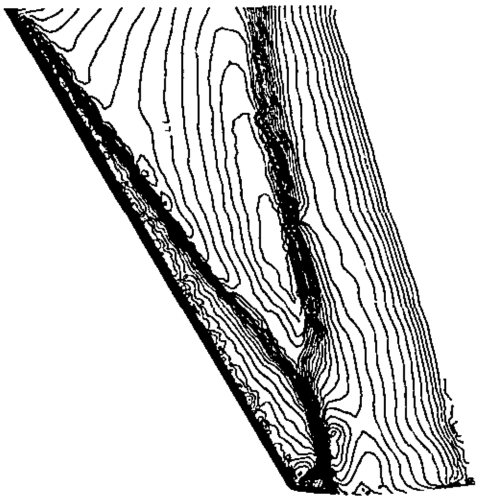


FIG. 16. Pressure coefficients comparison on the wing surface at $\eta = 0.65$ spanwise location: •, experimental values (upper surface); ◦, experimental values (lower surface); ———, one level adapted grid.

three spanwise locations. The values corresponding to the initial coarse grid calculation are shown in the short dashed lines in the figures. It is seen that the computed values are quite different from the experimental data at all spanwise locations and that there is no indication of a clear aft shock. The grid is now adapted using the same detection parameters as before. The once-adapted grid has 37,123 nodes and 206,577 cells. Figure 23 shows that the first level embedding (light shaded areas) covers a predominant portion of the wing surface. The solution obtained on the once-adapted grid is shown in Fig. 24. Comparing Figs. 19 and 24, it is observed that the fore shock is lot sharper and is very well captured. The aft shock is now captured well up to about $\eta = 15\,\%$, as the first level embedding extends only up to that region, beyond which the shock diffuses out. This can be attributed to the coarseness of the grid near the symmetry plane. Comparison of the pressure coefficients in Figs. 20 to 22 (long dashed lines) shows that there is a marked improvement in the results which now match the

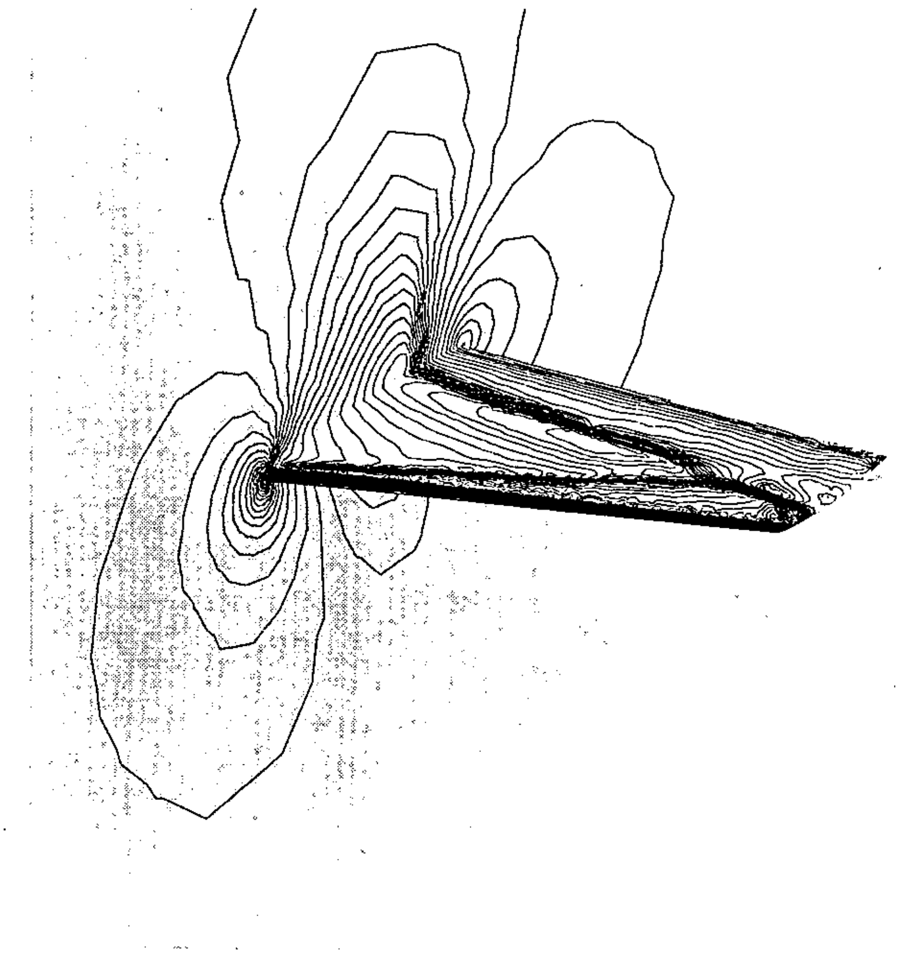


FIG. 17. Pressure coefficients comparison on the wing surface at $\eta = 0.90$ spanwise location: •, experimental values (upper surface); ○, experimental values (lower surface); ———, one level adapted grid.

experimental values quite closely at all spanwise locations of the wing. To improve the solution further, the grid is adapted again and the resulting second-level embedded grid on the wing surface is shown in Fig. 23 (darker shaded areas). The twice-adapted grid has 144,548 nodes and 833,701 cells. The second level covers the fore shock, the aft shock, as well as the leading edge region. It is observed that relatively large cells are next to much smaller ones, which makes this case a severe test for accuracy and robustness of the developed Euler solver. Mach number line contours in Fig. 25 show that the fore shock is very sharply captured. Comparing the pressure coefficient values on the wing surface (solid lines) in Figs. 20 to 22, it is seen that the fore shock is much sharper than the result obtained on the once-adapted fine grid. The aft shock is also considerably sharper than the result obtained on the once-adapted coarse grid, but little improvement is attained in the region beyond $\eta = 0.15$, as there is no embedding there.

9.2. *Low-Wing Transport Aircraft*

Computations were carried out for a low wing transport (LWT) configuration described in [27]. Inviscid, transonic flow of $M_\infty = 0.768$ was considered with an angle of attack $\alpha = 1.116°$. A semispan computational grid was generated for the LWT aircraft geometry without the nacelles. View of the triangulation on the wing upper surface and the fuselage is shown in Fig. 26. The initial grid is comprised of 48,828 nodes and 266,400 cells. The experimental pressure measurements were obtained with Reynolds number of 2.5×10^6 based on the mean aerodynamic chord of the wing [29].

Computed solutions obtained on the initial grid are shown in Fig. 27. Mach number contour lines are plotted at intervals of $\Delta M = 0.02$. A single shock wave is formed on the upper surface. Comparison of the pressure coefficients at three spanwise locations, namely $\eta = 0.20$, $\eta = 0.40$, and

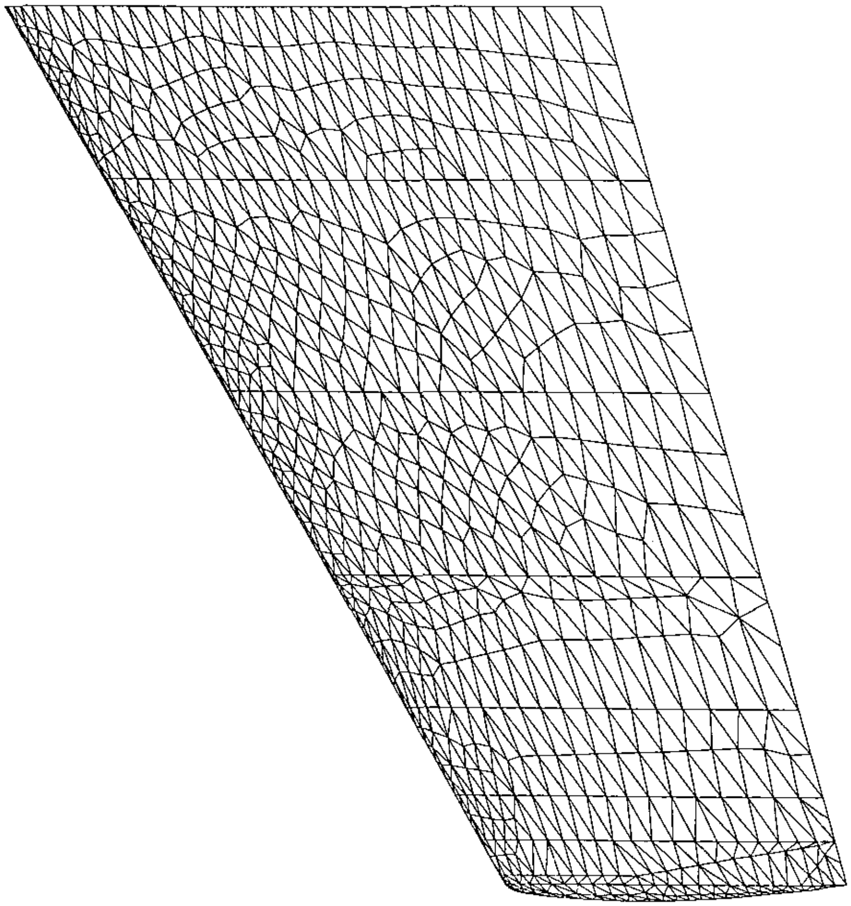
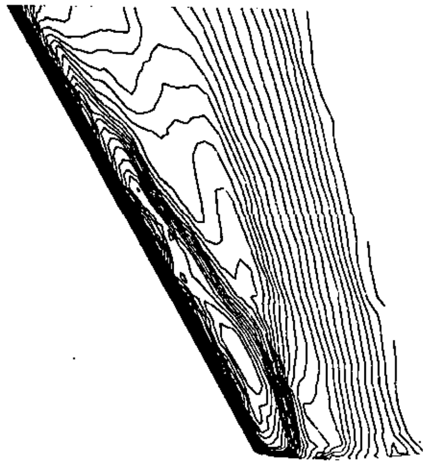


FIG. 18. Triangulation on the ONERA M6 wing upper surface corresponding to the initial coarse grid.

$\eta = 0.55$, on the wing surface are shown in Figs. 28 to 30. The experimental values corresponding to the upper surface are shown in filled circles and the values corresponding to the lower surface are shown in unfilled circles. Comparing the C_p values obtained on the initial grid, shown in dashed lines in Figs. 28 to 30, with the experimental data shows that



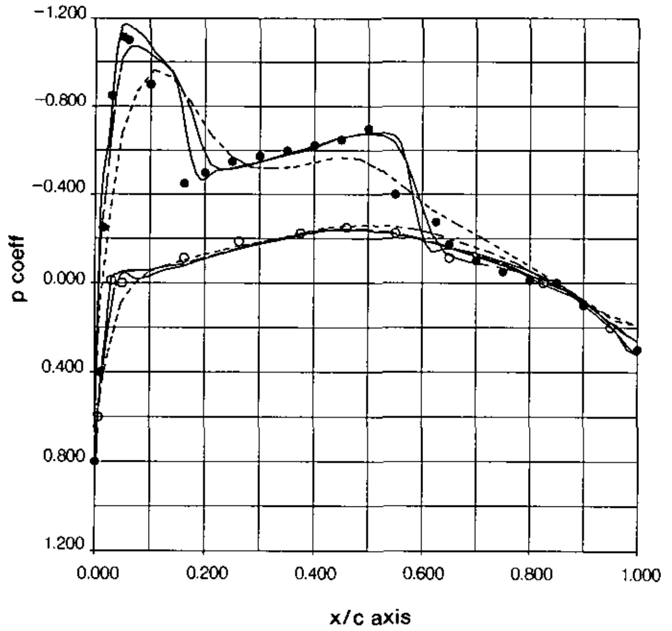


FIG. 20. Pressure coefficients comparison on the wing surface at $\eta = 0.44$ span-wise location: •, experimental values (upper surface);

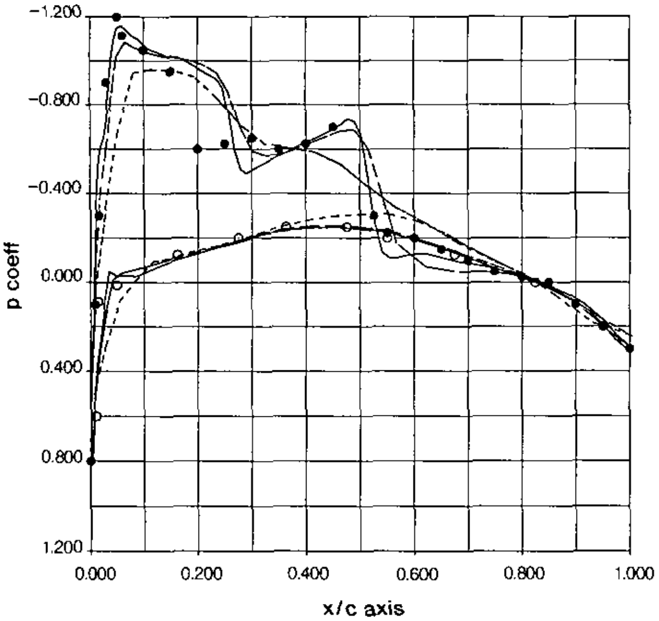


FIG. 21. Pressure coefficients comparison on the wing surface at $\eta = 0.65$ span-wise location: •, experimental values (upper surface); ○, experimental values (lower surface); ---, initial grid solution; ———, one level adapted grid; ———, two level adapted grid.

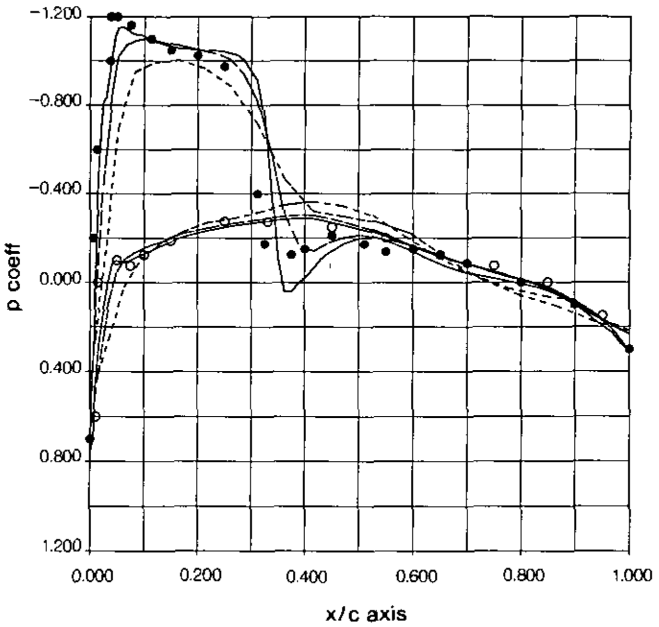


FIG. 22. Pressure coefficients comparison on the wing surface at $\eta = 0.90$ span-wise location: •, experimental values (upper surface); ○, experimental values (lower surface); ---, initial grid solution; ———, one level adapted grid; ———, two level adapted grid.

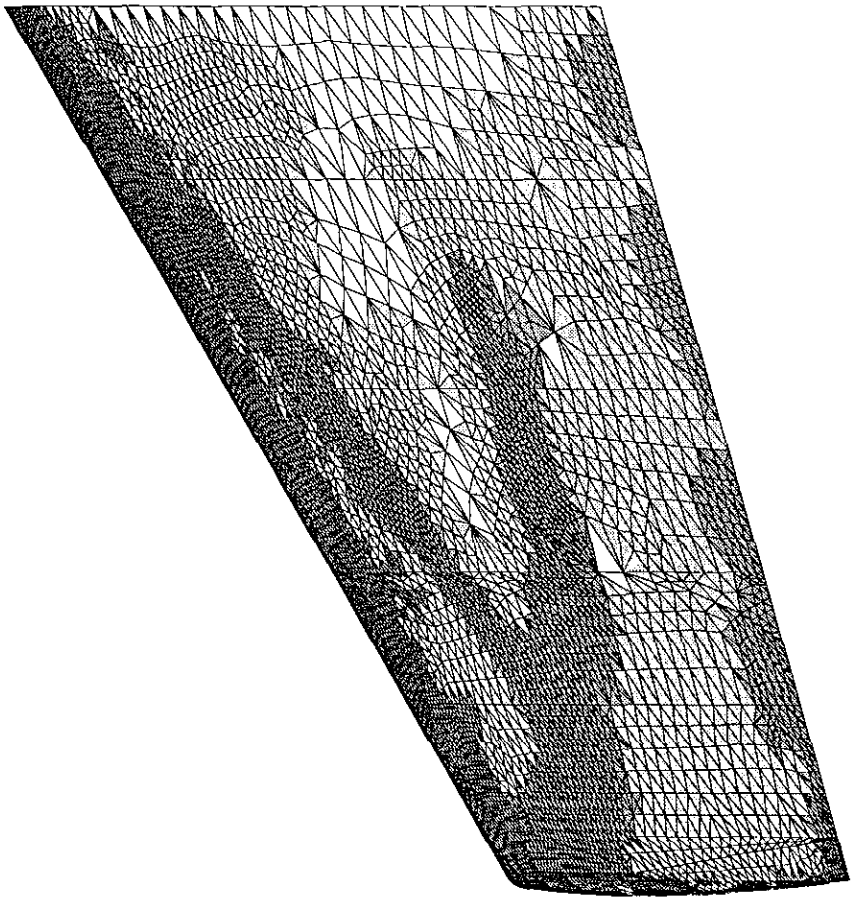


FIG. 23. Triangulation on the wing upper surface corresponding to the twice adapted coarse grid. Light shaded area denotes the first level and the dark shaded area denotes the second level.

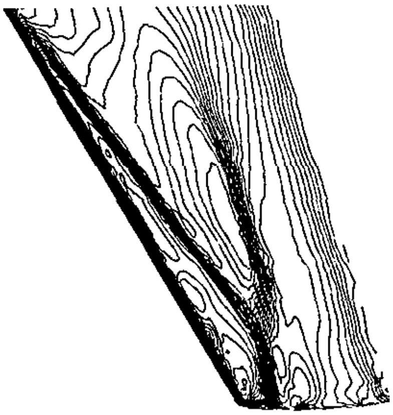


FIG. 24. Mach number contour lines on the wing upper surface. Solution obtained on the once adapted coarse grid ($\Delta M = 0.02$).

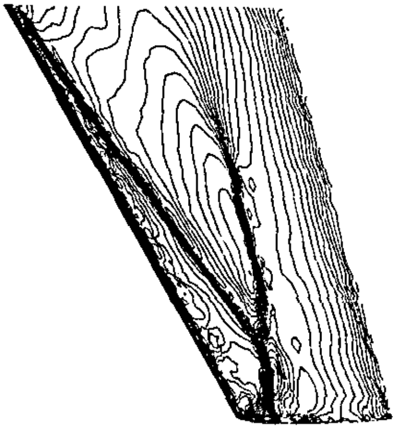


FIG. 25. Mach number contour lines on the wing upper surface. Solution obtained on the twice adapted coarse grid ($\Delta M = 0.02$).

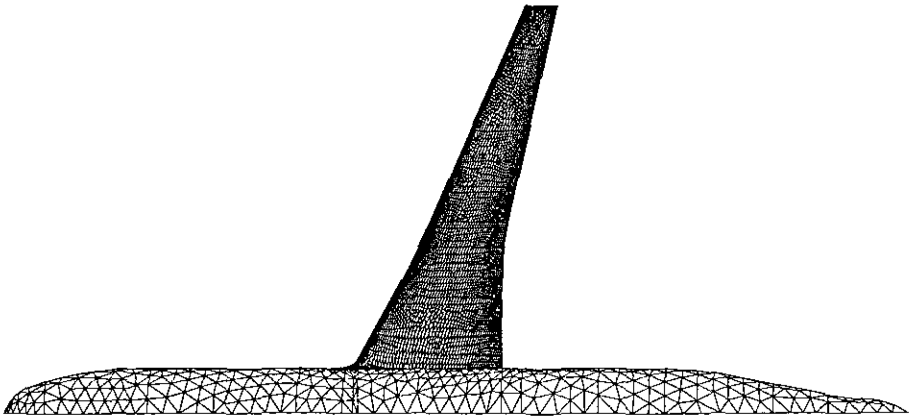


FIG. 26. Triangulation on the LWT aircraft wing and body surface corresponding to the initial grid.

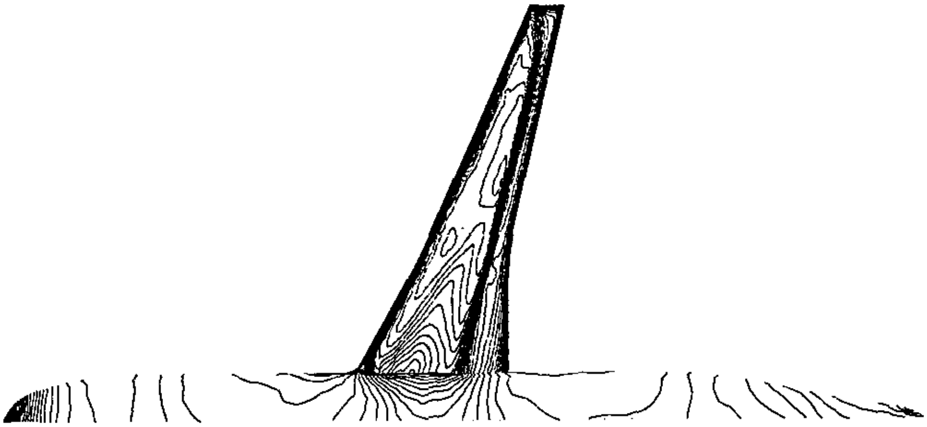


FIG. 27. Mach number contour lines on the wing upper surface. Solution obtained on the initial grid ($\Delta M = 0.02$).

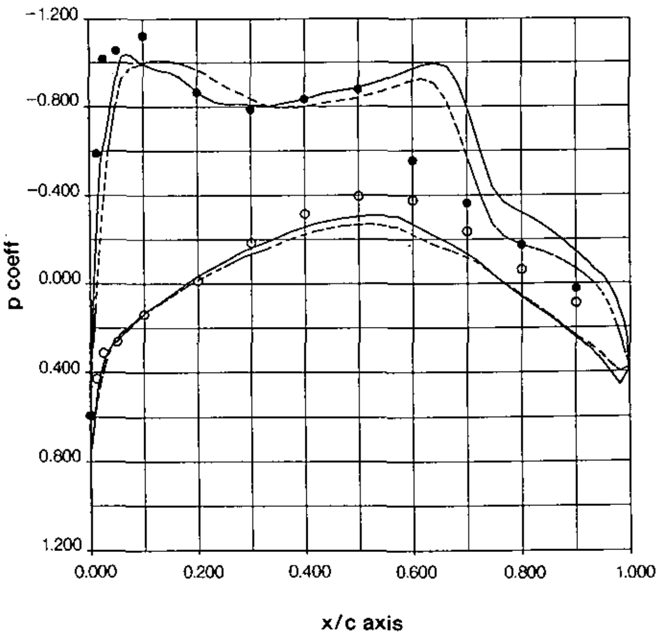


FIG. 28. Pressure coefficients comparison on the wing surface at $\eta = 0.20$ span-wise location: •, experimental values (upper surface); ∘, experimental values (lower surface); ---, initial grid solution; ———, one level adapted grid.

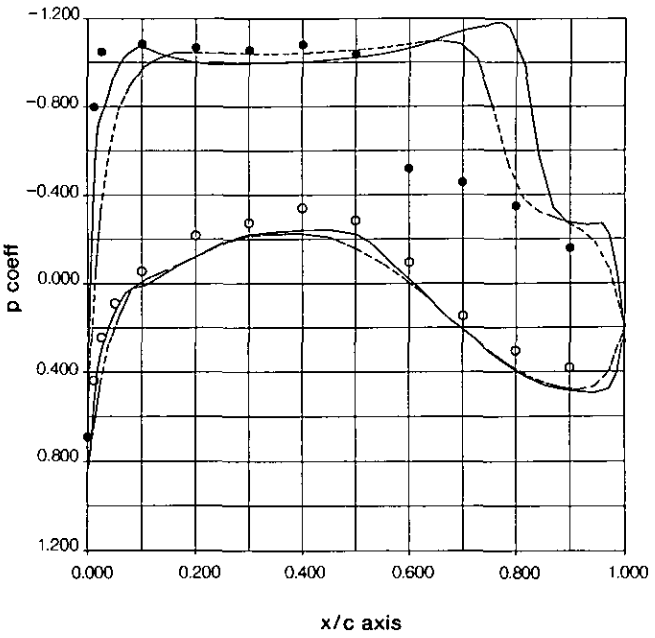


FIG. 30. Pressure coefficients comparison on the wing surface at $\eta = 0.55$ span-wise location: •, experimental values (upper surface); ∘, experimental values (lower surface); ---, initial grid solution; ———, one level adapted grid.

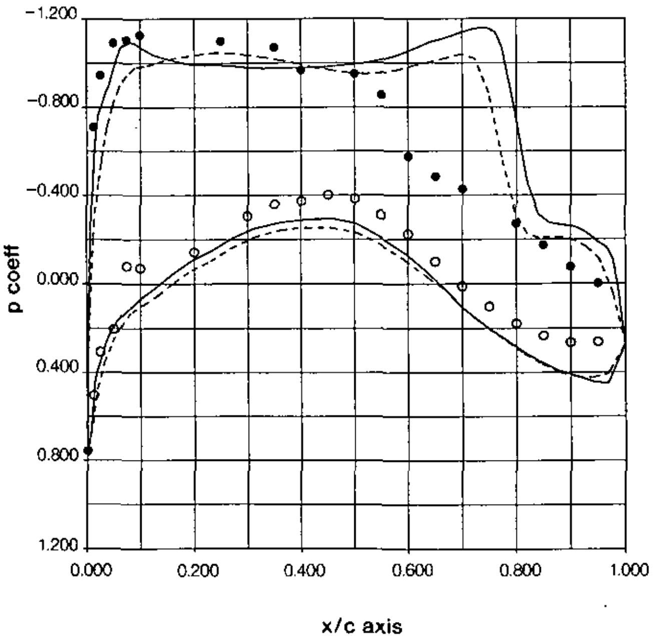


FIG. 29. Pressure coefficients comparison on the wing surface at $\eta = 0.40$ span-wise location: •, experimental values (upper surface); ∘, *experimental values (lower surface); ---, initial grid solution;* ———, one level adapted grid.

the results agree quite well in the fore regions of the wing, but agreement is somewhat poor in the aft regions. This has been observed by other inviscid flow computations as well [30]. The viscous effects change the flow pattern to a considerable extent in this case. Viscous effects have been observed to be very significant in the aft portion of the wing due to flow separation. The grid was embedded using the velocity gradients and velocity differences as the parameters. The embedded grid, on the aircraft surface as well as on the symmetry plane, is shown in Figs. 31 and 32. The adapted grid has 185,230 nodes. The Mach number contour lines are shown in Fig. 33, plotted at intervals of $\Delta M = 0.02$. The plot shows that there is a considerable improvement in the solution as compared to the one obtained on the initial grid. The shock appears distinctly sharper on the adapted grid solution. Comparison of the pressure coefficients, obtained using the adapted grid solution, on the wing surface with the experimental values is shown in solid lines in Figs. 28 to 30. The figures show that the solution on the adapted grid captures the section peak in the fore region of the wing better than the initial grid solution.

10. CONCLUDING REMARKS

A new Euler scheme for adaptive tetrahedral grids has been developed and applied. Accuracy of the method has been tested through comparisons with experimental data.

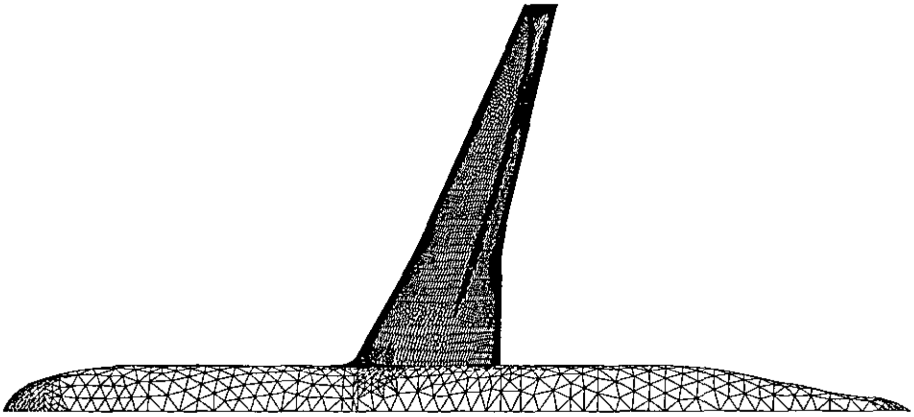


FIG. 31. Triangulation on the wing and body surface corresponding to the once adapted grid.

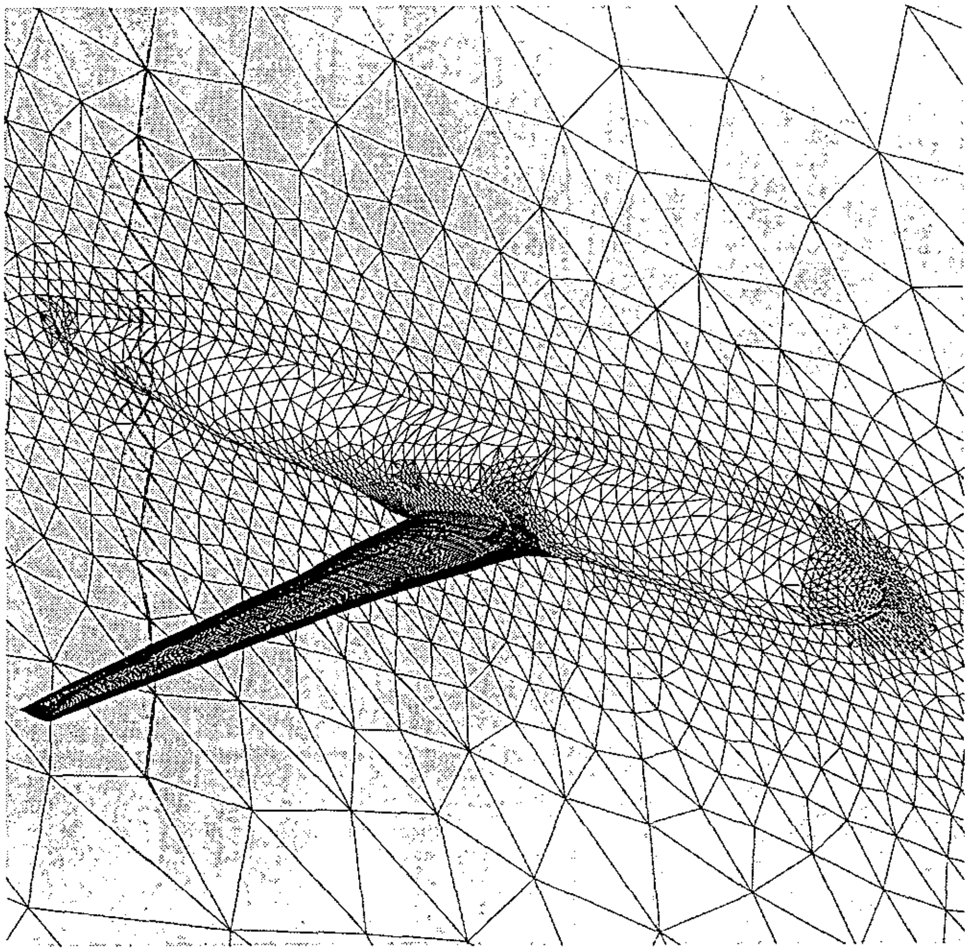


FIG. 32. Isometric view of the triangulation on the wing, body, and symmetry plane surface corresponding to the once adapted grid.

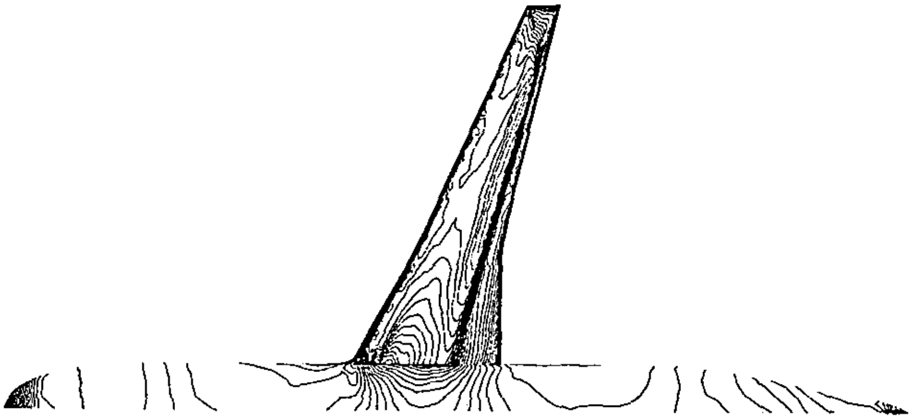


FIG. 33. Mach number contour lines on the wing upper surface. Solution obtained on the once adapted grid ($\Delta M = 0.02$).

The combined solver and grid-adapter gave relatively accurate solutions. The upwind-like smoothing operators yielded a stable scheme with grids that are quite non-uniform. Furthermore, they produced sharper shocks than other operators. The Lax–Wendroff-type of time integration proved to yield a stable scheme. Casting of the scheme into edge-based operations reduced computational effort by a factor of N, when compared to the classical cell-based approach, where N is the number of cell neighbors for an

7. J. F. Dannenhoffer III and J. R. Baron, AIAA Paper 85-0484, 1985 (unpublished).

8. J. T. Oden, T. Strouboulis, and P. Devloo, *Int. J. Num. Methods Fluids* 7 (1987), 1211.

9. Y. Kallinderis, Ph.D. thesis, MIT, Dept. of Aeronautics and Astronautics, CFDL Report No. TR-89-5, May 1989 (unpublised).

10. J. G. Kallinderis and J. R. Baron, *AIAA J.* **27** (1989), 37.

11. Y. Kallinderis, *AIAA J.* **30** (1992), 631.

12. Y. Kallinderis, *J. Comput. Phys.* **98** (1992), 129.

13. R. Lohner, AIAA Paper 87-0555, 1987 (unpublished).