AIAA-91-0102

A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes

Neal T. Frink†
NASA-Langley Research Center
Paresh Parikh‡
Shahyar Pirzadeh‡
ViGYAN, Inc.
Hampton, Virginia

Abstract

An upwind scheme is presented for solving the three-dimensional Euler equations on unstructured tetrahedral meshes. Spatial discretization is accomplished by a cell-centered finite-volume formulation using flux-difference splitting. Higher-order differences are formed by a novel cell reconstruction process which results in computational times per cell comparable to those of structured codes. The approach yields highly resolved solutions in regions of smooth flow while avoiding oscillations across shocks without explicit limiting. Solutions are advanced in time by a 3-stage Runge-Kutta time-stepping scheme with convergence accelerated to steady state by local time stepping and implicit residual smoothing.

Solutions are presented for a range of configurations in the transonic speed regime to demonstrate code accuracy, speed, and robustness. The results include an assessment of grid sensitivity and convergence acceleration by mesh sequencing.

Introduction

Solution algorithms for the Euler and Navier-Stokes equations on unstructured meshes have evolved rapidly in recent years, e.g. Refs. 1-12. Key motivations behind these developments include the need for more geometric flexibility in constructing quality meshes around complex configurations, a random data structure to better facilitate adapting the mesh to the physics of the flow, and a means to more easily accommodate moving meshes. Potential applications are many: more rapid analysis of full aircraft configurations, airframe component integration, iterative design, and store separation to name a few.

Most of the unstructured algorithms developed to date are based on either the finite element method or central differencing with added dissipation. Only in recent years has upwind differencing been investigated for unstructured grids[7-12]. Upwind differencing utilizes the propagation of information within a mesh in accordance with the theory of characteristics in constructing type-dependent differencing for components of the information traveling in opposite directions in a separate and stable manner. While this approach is more computationally intensive than central differencing, it does offer the advantages of being more robust and requiring less user interaction.

This paper describes a new upwind scheme which is analytically equivalent to that described in Ref. 12, but requires one-half the computational time. The scheme is a tetrahedral cell-centered, finite-volume upwind formulation using flux-difference splitting. Higher-order accuracy is achieved by a fast multidimensional linear reconstruction algorithm. The approach yields highly resolved solutions in regions of smooth flow while avoiding oscillations across shocks without explicitly applying a limiter. Solutions are advanced in time by a 3-stage Runge-Kutta time stepping scheme with convergence accelerated to steady state by local time stepping and implicit residual smoothing.

Governing Equations

The fluid motion is governed by the time dependent Euler equations for an ideal gas which express the conservation of mass, momentum, and energy for a compressible inviscid nonconducting adiabatic fluid in the absence of external forces. The equations are given below in integral form for a bounded domain Ω with a boundary $\partial\Omega$

$$\frac{\partial}{\partial t}\int\int\int_{\Omega}\mathbf{Q}dV + \int\int_{\partial\Omega}\mathbf{F}(\mathbf{Q})\cdot\hat{\mathbf{n}}dS = 0 \qquad (1)$$

where

$$\mathbf{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e_o \end{Bmatrix}$$

†Research Engineer, Senior member AIAA
‡Research Engineer, Member AIAA

and

$$\mathbf{F}(\mathbf{Q})\cdot\hat{\mathbf{n}} = (\mathbf{V}\cdot\hat{\mathbf{n}})\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ e_o+p\end{Bmatrix} + p\begin{Bmatrix}0\\ \hat{n}_x\\ \hat{n}_y\\ \hat{n}_z\\ 0\end{Bmatrix}$$

The equations are nondimensionalized with a reference density ρ_∞ and a speed of sound a_∞. Here $\hat{n}_x, \hat{n}_y$, and $\hat{n}_z$ are the Cartesian components of the exterior surface unit normal $\hat{\mathbf{n}}$ on the boundary $\partial\Omega$. The Cartesian velocity components are u, v, and w in the x, y, and z directions, respectively. The term e_o is the total energy per unit volume. With the ideal gas assumption, the pressure and total enthalpy can be expressed as

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

and

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

where γ is the ratio of specific heats and is prescribed as 1.4 for air.

Equation (1) describes a relationship where the time rate of change of the state vector $\mathbf{Q}$ within the domain Ω is balanced by the net flux $\mathbf{F}$ across the boundary surface $\partial\Omega$. The domain is divided into a finite number of tetrahedral cells, and Eq.(1) is applied to each cell. The state variables $\mathbf{Q}$ are volume-averaged values. It can be shown that the difference equations at each cell volume are satisfied exactly when evaluated at uniform freestream conditions.

Spatial Discretization

Flux Splitting

Flux quantities are computed using Roe's[13] flux-difference-splitting. The flux across each cell face κ is computed using Roe's numerical flux formula

$$\mathbf{F}_\kappa = {}^1\!/_2[\mathbf{F}(\mathbf{Q_L}) + \mathbf{F}(\mathbf{Q}_R) - \mid \tilde{\mathbf{A}} \mid (\mathbf{Q}_R - \mathbf{Q}_L)]_\kappa$$

Here $\mathbf{Q}_L$ and $\mathbf{Q}_R$ are the state variables to the left and right of the interface κ. The matrix $\tilde{\mathbf{A}}$ is computed from evaluating

$$\mathbf{A} \equiv \partial\mathbf{F}/\partial\mathbf{Q}$$

with Roe-averaged quantities such as:

$$\tilde{\rho} = \sqrt{\rho_L\rho_R}$$

$$\tilde{u} = (u_L + u_R\sqrt{\rho_R/\rho_L})/(1+\sqrt{\rho_R/\rho_L})$$

$$\tilde{v} = (v_L + v_R\sqrt{\rho_R/\rho_L})/(1+\sqrt{\rho_R/\rho_L})$$

$$\tilde{w} = (w_L + w_R\sqrt{\rho_R/\rho_L})/(1+\sqrt{\rho_R/\rho_L})$$

$$\tilde{h}_o = (h_{o_L} + h_{o_R}\sqrt{\rho_R/\rho_L})/(1+\sqrt{\rho_R/\rho_L})$$

$$\tilde{a}^2 = (\gamma-1)(\tilde{h}_o - (\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)/2)$$

so that

$$\mathbf{F}(\mathbf{Q}_R) - \mathbf{F}(\mathbf{Q}_L) = \tilde{\mathbf{A}}[\mathbf{Q}_R - \mathbf{Q}_L]$$

is satisfied exactly. Introducing the diagonalizing matrices $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{T}}^{-1}$, and the diagonal matrix of eigenvalues $\mathbf{\Lambda}$, then $\mid \tilde{\mathbf{A}} \mid$ is defined as

$$\mid \tilde{\mathbf{A}} \mid = \tilde{\mathbf{T}} \mid \tilde{\mathbf{\Lambda}} \mid \tilde{\mathbf{T}}^{-1}$$

The term

$$\mid \tilde{\mathbf{A}} \mid (\mathbf{Q}_R - \mathbf{Q}_L) = \tilde{\mathbf{T}} \mid \tilde{\mathbf{\Lambda}} \mid \tilde{\mathbf{T}}^{-1}\Delta\mathbf{Q}$$

in Roe's flux formula can be reduced to three $\Delta\mathbf{F}$ flux components, each of which is associated with a distinct eigenvalue:

$$\tilde{\mathbf{T}} \mid \tilde{\mathbf{\Lambda}} \mid \tilde{\mathbf{T}}^{-1}\Delta\mathbf{Q} = \mid \Delta\tilde{\mathbf{F}}_1 \mid + \mid \Delta\tilde{\mathbf{F}}_4 \mid + \mid \Delta\tilde{\mathbf{F}}_5 \mid$$

with

$$\mid \Delta\tilde{\mathbf{F}}_1 \mid = \mid \tilde{U} \mid \left\{ (\Delta\rho - \tfrac{\Delta p}{\tilde{a}^2}) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \frac{\tilde{u}^2+\tilde{v}^2+\tilde{w}^2}{2} \end{bmatrix} + \tilde{\rho} \begin{bmatrix} 0 \\ \Delta u - \hat{n}_x\Delta U \\ \Delta v - \hat{n}_y\Delta U \\ \Delta w - \hat{n}_z\Delta U \\ \tilde{u}\Delta u + \tilde{v}\Delta v + \tilde{w}\Delta w - \tilde{U}\Delta U \end{bmatrix} \right\}$$

$$\mid \Delta\tilde{\mathbf{F}}_{4,5} \mid = \mid \tilde{U} \pm \tilde{a} \mid (\frac{\Delta p \pm \tilde{\rho}\tilde{a}\Delta U}{2\tilde{a}^2}) \begin{bmatrix} 1 \\ \tilde{u} \pm \hat{n}_x\tilde{a} \\ \tilde{v} \pm \hat{n}_y\tilde{a} \\ \tilde{w} \pm \hat{n}_z\tilde{a} \\ \tilde{h}_o \pm \tilde{U}\tilde{a} \end{bmatrix}$$

where $\tilde{U} = \tilde{u}\hat{n}_x + \tilde{v}\hat{n}_y + \tilde{w}\hat{n}_z$ and $\Delta U = \hat{n}_x\Delta u + \hat{n}_y\Delta v + \hat{n}_z\Delta w$.

For a first-order scheme, the state of the primitive variables at each cell face is set to the cell-centered averages on either side of the face.

Modified Higher-Order Scheme

The challenge in constructing an effective higher-order scheme is to determine an accurate estimate of the left and right states at the cell faces for the flux calculation. Barth and Jespersen proposed a multidimensional linear reconstruction approach[7] which forms

the basis for the present scheme. In the cell reconstruction approach, higher-order accuracy is achieved by expanding the cell-centered solution to each cell face with a Taylor series:

$$\mathbf{q}(x,y,z) = \mathbf{q}(x_c,y_c,z_c) + \nabla\mathbf{q}_c \cdot \Delta\mathbf{r} \; + \; O(\Delta r^2) \quad (2)$$

where

$$\mathbf{q} \equiv [\rho, u, v, w, p]^T$$

This formulation requires that the solution gradient be known at the cell centers.

A new scheme was proposed in Ref. 12 for estimating the solution gradient. The general approach was to: 1) coalesce surrounding cell information to the vertices or nodes of the candidate cell, then 2) apply the mid-point trapezoidal rule to evaluate the surface integral of the gradient theorem

$$\nabla\mathbf{q}_c = \frac{1}{V_\Omega}\oint_{\partial\Omega} \mathbf{q}\hat{\mathbf{n}}dS \quad (3)$$

over the faces of each tetrahedral cell. Here, V_Ω denotes the volume enclosed by the surface Ω.

It is possible to further simplify the method of Ref. 12 such that Eq.(3) need not be evaluated explicitly. The simplification stems from some useful geometrical invariant features of triangles and tetrahedra. These features are illustrated for an arbitrary tetrahedral cell in Fig. 1. Note that a line extending from a cell-vertex through the cell-centroid will always intersect the centroid of the opposing face. Furthermore, the distance from the cell-vertex to the cell-centroid is always three-fourths of that from the vertex to the opposing face. (For a triangle, the comparable ratio of distance is two-thirds). By using these invariants along with the fact that Δr is the distance between the cell centroid and the face centroid, the second term in Eq.(2) can be evaluated as:

$$\nabla\mathbf{q}_c \cdot \Delta\mathbf{r} = \frac{\partial\mathbf{q}}{\partial r}\Delta r$$

$$\approx \left[\frac{1/3(\mathbf{q}_{n_1} + \mathbf{q}_{n_2} + \mathbf{q}_{n_3}) - \mathbf{q}_{n_4}}{4\Delta r}\right]\Delta r \quad (4)$$

Thus, Eq.(2) can be approximated for tetrahedral cells by the simple formula:

$$\mathbf{q}_{f_{1,2,3}} = \mathbf{q}_c + 1/4\,[1/3(\mathbf{q}_{n_1} + \mathbf{q}_{n_2} + \mathbf{q}_{n_3}) - \mathbf{q}_{n_4}] \quad (5)$$

where as illustrated in Fig. 1, the subscripts n_1, n_2, n_3 denote the nodes comprising face $f_{1,2,3}$ of cell c and n_4 corresponds to the opposite node. This modified scheme is analytically equivalent to that in Ref. 12 and results in a factor of two reduction in computational time of the flow solver.

The nodal quantities $\mathbf{q}_n$ are determined in the manner described in Ref. 12. Accordingly, estimates of the solution are determined at each node by a weighted average of the surrounding cell-centered solution quantities. It is assumed in the nodal averaging procedure that the known values of the solution are concentrated at the cell centers, and that the contribution to a node from the surrounding cells is inversely proportional to the distance from each cell centroid to the node:

$$\mathbf{q}_n = \left(\sum_{i=1}^{N}\frac{\mathbf{q}_{c,i}}{r_i}\right)/\left(\sum_{i=1}^{N}\frac{1}{r_i}\right) \quad (6)$$

where

$$r_i = [(x_{c,i} - x_n)^2 + (y_{c,i} - y_n)^2 + (z_{c,i} - z_n)^2]^{\frac{1}{2}}$$

The subscripts n and c,i refer to the node and surrounding cell-centered values, respectively. Note that the reconstruction process utilizes information from all of the cells surrounding the candidate cell, thus producing a truly multidimensional higher-order expansion in Eq.(5). For boundary nodes, the surrounding face-centered boundary conditions and respective distances are used in Eq.(6).

Upwind schemes generally require the use of limiter functions to obtain smooth higher-order solutions around flow discontinuities, which has been the experience with the schemes of Refs. 7-11. While the present method has yet to be applied in two-dimensions, experience with its application in three-dimensions has shown that limiting is not required, and that the method correctly captures shocks without oscillations. While this unexpected result is beneficial, it seems unlikely that a high-order scheme could capture discontinuities smoothly without some form of artificial dissipation being added. It is quite possible that the particular averaging procedure employed in the present algorithm could add dissipation and locally reduce the accuracy across discontinuities. With respect to limiting overshoots in the expansion of Eq.(5), it can be reasoned that when the averaging procedure of Eq.(6) is applied at a node, the resulting $\mathbf{q}_n$ represents a weighted mean value of the surrounding solution, i.e. $\mathbf{q}_n$ is bounded by the extrema of the surrounding solution. Furthermore, in three dimensions the summation of Eq.(6) accesses an average of 20 to 22 cells for each node, which results in a smoothing of errors introduced from the surrounding solution. Thus, the expansion will have been smoothed and bounded by the procedure and should not introduce new extrema into the solution.

The accuracy of the higher-order scheme has not been formally determined. It is known that the Taylor series expansion and the midpoint-trapezoidal rule,

which was applied explicitly in Ref. 12 and implicitly in the present algorithm, are both second-order accurate. Thus, the key to accuracy for this method lies in the quality of the averaged solution at the nodes. Several averaging procedures were investigated with known test functions, both linear and nonlinear, on an arbitrary tetrahedral grid. Among the alternate procedures were those based on cell volumes, and on the inverse distances from cell-center to node raised to a power. Error was assessed by an RMS average of the local errors relative to the exact solution. Volume averaging yielded the poorest result, while averaging based on the inverse distance raised to the first power (Eq.(6)) worked best.

Time Integration

A semidiscrete form of the governing equations reads

$$V_i \frac{\partial \mathbf{Q}_i}{\partial t} + \mathbf{R}_i = \mathbf{0}, \quad i = 1, 2, 3, \ldots \tag{6}$$

where

$$\mathbf{R}_i = \sum_{j=\kappa(i)} \mathbf{F}_{i,j} \Delta S_{i,j}$$

R_i is the residual accrued by summation of the fluxes through the four faces κ of a tetrahedral cell i. These equations are integrated in time using a fully explicit m-stage Runge-Kutta time-stepping scheme developed by Jameson et al.[14]:

$$\begin{aligned}
\mathbf{Q}_i^{(0)} &= \mathbf{Q}_i^n \\
\mathbf{Q}_i^{(1)} &= \mathbf{Q}_i^{(0)} - \alpha_1 \frac{\Delta t}{V_i} \mathbf{R}_i^{(0)} \\
&\cdots \\
\mathbf{Q}_i^{(m-1)} &= \mathbf{Q}_i^{(0)} - \alpha_{m-1} \frac{\Delta t}{V_i} \mathbf{R}_i^{(m-2)} \\
\mathbf{Q}_i^{(m)} &= \mathbf{Q}_i^{(0)} - \alpha_m \frac{\Delta t}{V_i} \mathbf{R}_i^{(m-1)} \\
\mathbf{Q}_i^{n+1} &= \mathbf{Q}_i^{(m)}
\end{aligned} \tag{7}$$

where the superscript n denotes the time level, and the parenthetical superscripts the stage of the Runge-Kutta time stepping. The weighting factors α_1 to α_m are defined as:

$$\alpha_k = \frac{1}{m-k+1}, \quad k = 1, \ldots, m$$

These values of α_k will give m-order accuracy in time for a linear equation. Preliminary calculations were made using both a 3-stage and 4-stage scheme. The solution and convergence characteristics were essentially identical. Thus, a 3-stage scheme was used for the calculations presented in this paper.

In many cases, time accuracy in the integration is not required. For such cases, the solution convergence to steady state is accelerated by local time stepping and implicit residual smoothing.

Local Time Stepping

Local time stepping accelerates convergence by advancing the solution at each cell in time at a CFL number near the local stability limit. The expression for the local time step was derived with the aid of a 2-D stability analysis presented in Ref. 9:

$$\Delta t_i \leq \nu \frac{V_i}{A_i + B_i + C_i} \tag{8}$$

with

$$\begin{aligned}
A_i &= (\mid u_i \mid + a_i) S_i^{(x)} \\
B_i &= (\mid v_i \mid + a_i) S_i^{(y)} \\
C_i &= (\mid w_i \mid + a_i) S_i^{(z)}
\end{aligned}$$

where ν is the CFL number, V_i is the cell volume, a_i is the local speed of sound, and $S_i^{(x)}$, $S_i^{(y)}$, and $S_i^{(z)}$ are the projected areas of cell i in the x, y, and z directions. The local time steps were updated every 50 cycles for the results presented in this paper.

Implicit Residual Smoothing

The maximum time step can be further increased by increasing the support of the scheme through implicit averaging of the residuals[15] with their neighbors. The residuals are filtered through a smoothing operator (which is essentially the Laplacian operator for a uniform grid):

$$\overline{\mathbf{R}_i} = \mathbf{R}_i + \varepsilon \nabla^2 \overline{\mathbf{R}_i}$$

where

$$\nabla^2 \overline{\mathbf{R}_i} = \sum_{j=\kappa(i)} (\overline{\mathbf{R}_j} - \overline{\mathbf{R}_i})$$

The summation uses residuals from the neighboring cells which share the faces κ with cell i. The resulting set of equations can easily be solved by using Jacobi iteration

$$\overline{\mathbf{R}_i}^{(m)} = \left(\mathbf{R}_i + \varepsilon \sum_{j=\kappa(i)} \overline{\mathbf{R}_j}^{(m-1)}\right) / \left(1 + \varepsilon \sum_{j=\kappa(i)} 1\right) \tag{9}$$

A value for ε of about 0.5 is suggested in Ref. 16 to maintain a strongly diagonally dominant coefficient matrix. In practice, two Jacobi iterations are adequate to give a good approximation of $\overline{\mathbf{R}_i}$ at all cell centers. Residual smoothing was performed during every stage of the Runge-Kutta time cycle and resulted in a doubling of the time step.

Boundary Conditions

For the solid boundaries such as the wing and centerplane, the flow tangency condition is imposed by setting the velocities on the boundary faces to their cell-center values and then subtracting the component normal to the solid surface. Density and pressure boundary conditions are simply set to the cell-centered value. A condition of zero mass and energy flux through the surface is ensured by setting the left and right states of solid boundary faces equal to the boundary conditions prior to computing the fluxes with Roe's approximate Riemann solver. This technique only permits a flux of the pressure terms of the momentum equations through a solid boundary.

Characteristic boundary conditions are applied to the far-field subsonic boundary using the fixed and extrapolated Riemann invariants corresponding to the incoming and outgoing waves. The incoming Riemann invariant is determined from the freestream flow and the outgoing invariant is extrapolated from the interior domain. The invariants are used to determine the locally normal velocity component and speed of sound. At an outflow boundary, the two tangential velocity components and the entropy are extrapolated from the interior, while at an inflow boundary they are specified as having far-field values. These five quantities provide a complete definition of the flow in the far field.

Results

A range of results are presented in this section to show the speed, accuracy and robustness of the flow solver. The speed and the accuracy is shown by way of grid sensitivity and mesh sequencing studies while its robustness is established by flow solutions on two complex three-dimensional configurations. All the meshes for this study were generated using an improved version of the advancing front grid generation program, VGRID3D [17].

ONERA M6 Wing

An ONERA M6 wing has been used for the grid sensitivity and the mesh sequencing studies. This configuration has been widely used as a benchmark to evaluate performance of newly developed flow solution methods. The wing has a leading edge sweep of 30 degrees, 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 and a semispan b of 1.0 with a rounded tip. The computational domain is bounded by a rectangular box with boundaries at $-6.5 \leq x \leq 11.0$, $0.0 \leq y \leq 2.5$, and $-6.5 \leq z \leq 6.5$.

Grid Sensitivity. Transonic solutions were computed on three grids (Fig. 2) at the same conditions: $M_\infty = 0.84$, and $\alpha = 3.06°$, to make an assessment of the grid sensitivity. One of the meshes (Mesh 1) has cells stretched in the spanwise direction where gradients are small while the other two meshes have no stretching. Mesh size specifications are listed in Table 1.

	Mesh 1	Mesh 2	Mesh 3
Total Cells	35008	108755	231507
Boundary Faces	4046	9858	16984
Total Nodes	6910	20412	42410
Boundary Nodes	2025	4931	8494

Table 1. Mesh size specifications.

The computations were performed using the 3-stage Runge-Kutta time stepping scheme with local time stepping, implicit residual smoothing, and a CFL number of 4.0. The solutions were started from freestream initial conditions with the first-order scheme and run until the L_2-norm (RMS average of all residuals) decreased one order of magnitude, at which time the solver automatically switched to the higher-order scheme. The solution history is plotted in Fig. 3 against CRAY-2S CPU time to provide a relative comparison of computational effort. Figure 3(a) shows the L_2-norm with a decrease of approximately 2.5 orders of magnitude. The convergence history of the lift coefficient is shown in Fig. 3(b). Additional details of the solution characteristics are provided in Table 2. For comparison, computations presented in Ref. 12 for Meshes 2 and 3 used approximately 3 hours and 8 hours of CRAY-2S CPU time, respectively.

	Dimensioned Memory, mw	CRAY-2S Time, min.	Number Cycles
Mesh 1	2.6	14	800
Mesh 2	8.1	89	1565
Mesh 3	17.1	203	1716

Table 2. Solution characteristics.

A comparison of wing surface pressure contours for the three meshes is presented in Fig. 4 with contour intervals of $\Delta(p/p_\infty) = 0.02$. Before plotting, the computed face-centered boundary quantities were averaged to the boundary nodes using Eq.(6). The contour results show very little overall sensitivity to mesh size. As expected, the primary effect of the grid occurs with the spatial resolution of the shocks.

Figure 5 shows the effect of mesh size on the streamwise surface C_p distribution at six span stations. The present results are plotted in comparison to experimental data at a Reynolds number of 11.7 million[18]. The computations on all the three meshes agree well

with experiment and demonstrate the accuracy of the calculations. The primary effect of mesh size is confined to regions of large gradients such as the leading-edge suction peak and the shock, where the finer mesh yields sharper shock definition. Comparisons shown in Ref. 12 demonstrated that the solutions for Meshes 2 and 3 are comparable to those obtained with structured codes. Although the deterioration of the solution for Mesh 1 is greater at the root and tip stations, it should be noted that its solution was obtained with an order of magnitude less CPU time and over 6 times less memory than that for Mesh 3.

The force and moment coefficients listed in Table 3 were computed by integrating the face-centered boundary pressures. The coefficients for lift, drag, pitching moment, and wing root bending moment are based on reference quantities of $S_{ref} = .5255$, $\bar{c} = .67$, and $b_{ref} = 1.0$. The pitching moment is referenced about the wing apex.

	Mesh 1	Mesh 2	Mesh 3
C_L	.2816	.2904	.2911
C_D	.0141	.0132	.0123
C_m	-.1688	-.1724	-.1726
C_{RBM}	.1270	.1283	.1285

Table 3. Force and moment coefficients.

Mesh Sequencing. A solution for a given configuration can be converged to steady state more efficiently by resorting to mesh sequencing. This procedure is a convergence acceleration technique in which the converged flow solution obtained on a coarse mesh is used as a starting solution for a finer mesh instead of starting from the uniform flow. The final steady state solution can be obtained in less overall computational time compared to one started from freestream initial conditions. Of course, interpolation of data between two completely different unstructured three-dimensional grids is not straight forward. A program which makes use of efficient octree data structures was written to accomplish this task.

Mesh sequencing was applied to the M6 wing to quantify the benefits in terms of reduced computational time. The converged solution from the stretched mesh (Mesh 1), was interpolated onto the fine mesh (Mesh 3) as a starting solution. Figure 6 shows the convergence history of the L_2-norm. To achieve the same level of residual reduction of approximately 2.5 orders of magnitude, the mesh sequenced solution results in a savings of 32-percent in CPU time. Even with accounting for the interpolation time, the net decrease in CPU time is still 29-percent. The final results are identical with and without mesh sequencing and, thus, are not shown.

Low-Wing Transport

Another quantitative assessment of the flow solver is made using a low-wing transport configuration described in Ref. 19. The 1/17th-scale configuration contains a supercritical airfoil and a flow-through representation of an advanced turbofan nacelle with a bypass ratio of approximately 6. The experimental pressure measurements were obtained in the NASA Langley 16-Foot Transonic tunnel[20] at transonic Mach numbers with Reynolds numbers in the range of 2.5 x 10^6 based on the mean aerodynamic chord of the wing. The present calculations were made for the condition of $M_\infty = 0.768$ and $\alpha = 1.116°$.

The computational grid consists of 418,939 cells and 75,470 total nodes representing the semispan configuration. The surface grid (Fig. 7) contains 21,428 boundary faces and 10,716 boundary nodes, including the outer boundaries and plane of symmetry. Grid stretching was not applied for these preliminary calculations. A sufficient definition of the internal flow-through nacelle geometry was not available, so the grid was terminated at a plane inside the inlet, and at the two bypass exit planes. Freestream conditions were prescribed on the exit plane boundaries. A condition of $M = 0.632M_\infty$ was imposed on the inlet plane to balance the mass flux.

Surface pressure contours are presented in Fig. 8. Good resolution of the wing shock can be observed along with evidence of an inboard lambda shock.

A comparison of the streamwise C_p distributions are shown in Fig. 9 for six span stations. The inviscid results are compared with experimental data at a slightly higher angle of attack. In general, the agreement is good and consistent with the expected effects of viscosity. The shock is more aft, and the effects of flow separation downstream of the shock and in the lower-surface cusp region result in a lower experimental ΔC_p over the aft region. (It should be noted that the stations within $0.463 \leq \eta \leq .70$ had only three cells defining the region between the shock and the trailing edge.) Similar comparisons are presented in Ref. 21 for a typical transport configuration with a supercritical wing. Comparisons shown between viscous and inviscid calculations and transonic experimental data show viscous effects comparable in magnitude to those observed in the present calculations.

The solution was obtained using mesh sequencing and required approximately 6 hours of CPU time and 31 megawords of memory. The present grid has a considerable number of cells clustered in directions of small gradients where they are not needed. The total number of cells could be greatly reduced with grid stretching, which would significantly decrease the amount of memory and CPU time required to obtain a satisfac-

tory solution. Generation of stretched grids for complex configurations is presently an active area of research.

Space Transportation System (STS)

A computation was made on the Space Transportation System (STS) at $M_\infty = 1.05$, and $\alpha = -3.1°$ to demonstrate the robustness of the flow solver in obtaining a solution on a complex geometry with a complex flow field. The semispan grid, which includes the orbiter, external tank, and solid rocket boosters, consists of 108,538 cells and 21,562 total nodes. The surface grid shown in Fig. 10 is represented by 13,552 triangular faces and 6,780 nodes, including the outer computational boundaries and plane of symmetry. The computations were made with zero elevon deflection.

The solution was obtained with 3410 first-order cycles with a CFL number of 0.5, then 1275 additional cycles at higher-order with a CFL number of 1.0. The solution required 235 minutes of CPU time on a CRAY-2S and used 8.2 megawords of memory.

Figure 10 shows a composite picture of the surface triangulation and the corresponding pressure contours on the full configuration. The centerplane grid is shown in Fig. 11 along with the pressure contours in Fig. 12. The basic features of the flow (shocks, expansions etc.) have been well captured in the solution considering that only 3-4 layers of fine cells have been used close to the body. The quality of the present results, which were computed on a relatively coarse grid with a relatively small amount of computer time, serves as a good demonstration of the flow solver capabilities.

Code Efficiency

The preceding solutions were computed on a single processor of the NASA Langley Research Center *Voyager* CRAY-2S. All coding within the flow solver portion of the code vectorizes with the standard FORTRAN compiler. The code statistics are summarized in Table 4.

Dimensioned Memory	74 words/cell
CPU Time, First Order	18 μs/cell/cycle
CPU Time, Higher Order	34 μs/cell/cycle

Table 4. Code statistics.

For comparison, the prior higher-order algorithm of Ref. 12 required 65 μs/cell/cycle.

To put these statistics in perspective, structured Euler codes generally require from 40 to 50 words of memory per cell, and 25 to 35 microseconds of CRAY-2S time per cell per cycle for higher-order solutions. Thus, the new algorithm yields efficiencies which are comparable to those of structured codes.

To avoid possible confusion, it should be noted that a 3-D structured mesh of hexahedral elements contains the same number of nodes as cells (asymptotically), whereas an unstructured mesh of tetrahedral elements generally contains between 5 and 6 times more cells than nodes. As shown in Ref. 12, comparable accuracy can be achieved between structured and cell-centered unstructured codes with the number of cells being of the same order of magnitude. Thus, it is important to make comparisons based on the number of unknowns computed, i.e. the number of cells for a cell-centered scheme.

Concluding Remarks

An upwind scheme for solving the three-dimensional Euler equations on unstructured tetrahedral meshes has been presented. The algorithm consists of a time-explicit cell-centered finite-volume formulation using flux-difference splitting. Higher-order accuracy is achieved by a fast multidimensional linear reconstruction algorithm which produces a computational efficiency comparable to that of structured algorithms. The approach yields highly resolved solutions in regions of smooth flow while avoiding oscillations across shocks without explicitly applying a limiter.

Results have been presented for a range of configurations at transonic speeds to demonstrate the speed, accuracy, and robustness of the flow solver. The potential for computational efficiency and accuracy has been illustrated by obtaining reasonably good solutions on the ONERA M6 wing in 14 minutes of CRAY-2S run time with 2.6 megawords of memory. A quantitative assessment has been presented for a low-wing transport configuration to demonstrated the robustness of the flow solver in providing an accurate solution on a complex geometry. A computation has been made on the Space Transportation System at a transonic Mach number to demonstrate the robustness of the flow solver in obtaining a solution on a very complex geometry with a complex flow field.

Acknowledgements

Development of the basic flow solver was undertaken in partial fulfillment of PhD requirements for the first author through VPISU. The author wishes to thank his advisor, Prof. R. W. Walters, for many stimulating discussions.

References

[1] Mavriplis, D.J.: Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes. *AIAA Journal*, Vol. 28, No. 2, Feb. 1990, pp. 213-221.

[2] Mallet, M.; Perieux, P.; Perrier, P.;, and Stoufflet, B.: Flow Modelization and Computational Methodologies for the Aerothermal Design of Hypersonic Vehicles: Application to European Hermes. AIAA Paper 88-2628, Jan. 1988.

[3] Desideri, J.A.; and Dervieux, A.: Compressible Flow Solvers Using Unstructured Grids, VKI Lecture Series, 1988-05.

[4] Rostand, P.; and Stoufflet, B.: TVD Schemes to Compute Compressible Viscous Flows on Unstructured Meshes. *Nonlinear Hyperbolic Equations*, Josef Ballmann and Rolf Jeltsch, eds., F. Vieweg, Publisher, Braunschweig, FRG, 1989, pp. 510-520.

[5] Marcum, D. L.; and Agarwal, R. K.: A Three-Dimensional Finite Element Navier-Stokes Solver with k-ε Turbulence Model for Unstructured Grids. AIAA Paper 90-1652, 1990.

[6] Vassberg, J. C.; and Dailey, K. B.: AIRPLANE: Experiences, Benchmarks, and Improvements. AIAA Paper 90-2998, 1990.

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

[8] Venkatakrishnan, V.; and Barth, T. J.: Application of Direct Solvers to Unstructured Meshes for the Euler and Navier-Stokes Equations Using Upwind Schemes. AIAA Paper 89-0364, 1989.

[9] Whitaker, D. L.: Two-Dimensional Euler Computations on a Triangular Mesh Using an Upwind, Finite-Volume Scheme. Ph.D Dissertation, Virginia Polytechnic Institute and State University, 1988.

[10] Batina, J. T.: Three-Dimensional Flux-Split Euler Schemes Involving Unstructured Dynamic Meshes. AIAA Paper 90-1649, 1990.

[11] Wey, T. C.; and Li, C. P.: Numerical Simulation of Shuttle Ascent Transonic Flow Using An Unstructured Grid Approach. Presented at "Symposium on Computational Technology for Flight Vehicles", Washington, D.C. Nov. 5-7, 1990.

[12] Frink, N. T.: Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes. Presented at workshop on "Accuracy of Unstructured Grid Techniques" held at NASA Langley Research Center, Jan. 16-17, 1990. Also accepted for publication in *AIAA Journal.*

[13] Roe, P. L.: Characteristic Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337-365.

[14] Jameson, A.; Schmidt, W.; and Turkel, E.: Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes. AIAA Paper 81-1259, June 1981.

[15] Jameson, A.; and Baker, T. J.: Solution of the Euler Equations for Complex Configurations. AIAA 83-1929-CP, July 1983.

[16] Mavriplis, D.: Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes. *AIAA Journal*, Vol. 26, July 1988, pp. 824-831.

[17] Parikh, P.; Pirzadeh, S.; and Löhner, R.: A Package for 3-D Unstructured Grid Generation, Finite-Element Flow Solution and Flow Field Visualization. NASA CR-182090, Sept. 1990.

[18] Schmitt, V.; and Charpin, F.: Pressure Distributions on the ONERA M6-Wing at Transonic Mach Number. AGARD Advisory Report 138, May 1979.

[19] Pendergraft, O. C.; Re, R. J.; and Kariya, T. T.: Nacelle/Pylon Interference Study on a 1/17th-Scale, Twin-Engine, Low-Wing Transport Model. AIAA Paper 89-2480, July, 1989.

[20] Corson, B. W., Jr.; Runckel, J. F.; and Igoe, W. B.: Calibration of the Langley 16-Foot Transonic Tunnel With Test Section Air Removal. NASA TR R-423, 1974.

[21] Melson, N. D.; and Streett, C. L.: Updated User's Guide for TAWFIVE With Multigrid. NASA TM-4109, 1989.

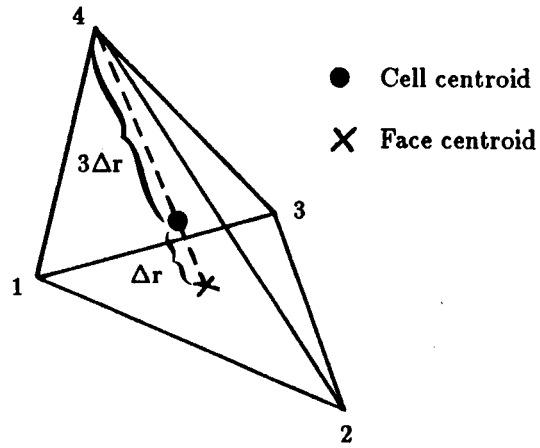
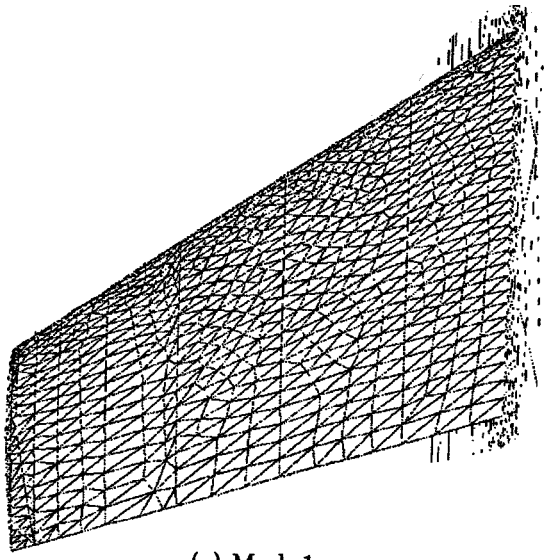
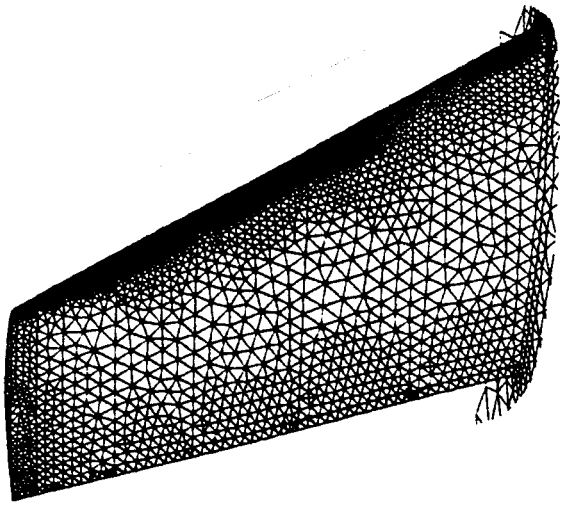


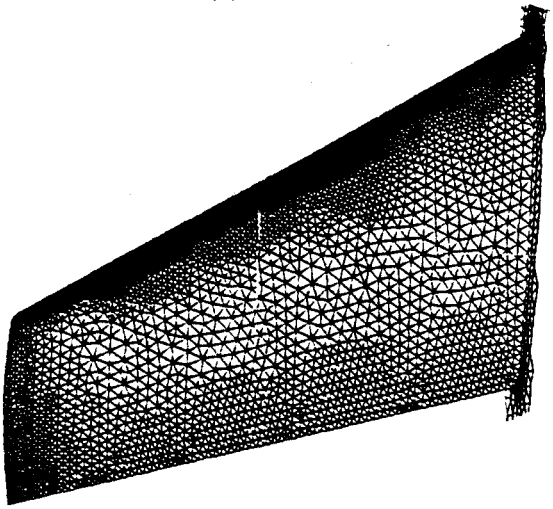
Figure 1.- Geometrically invariant features of tetrahedra.



(a) Mesh 1.

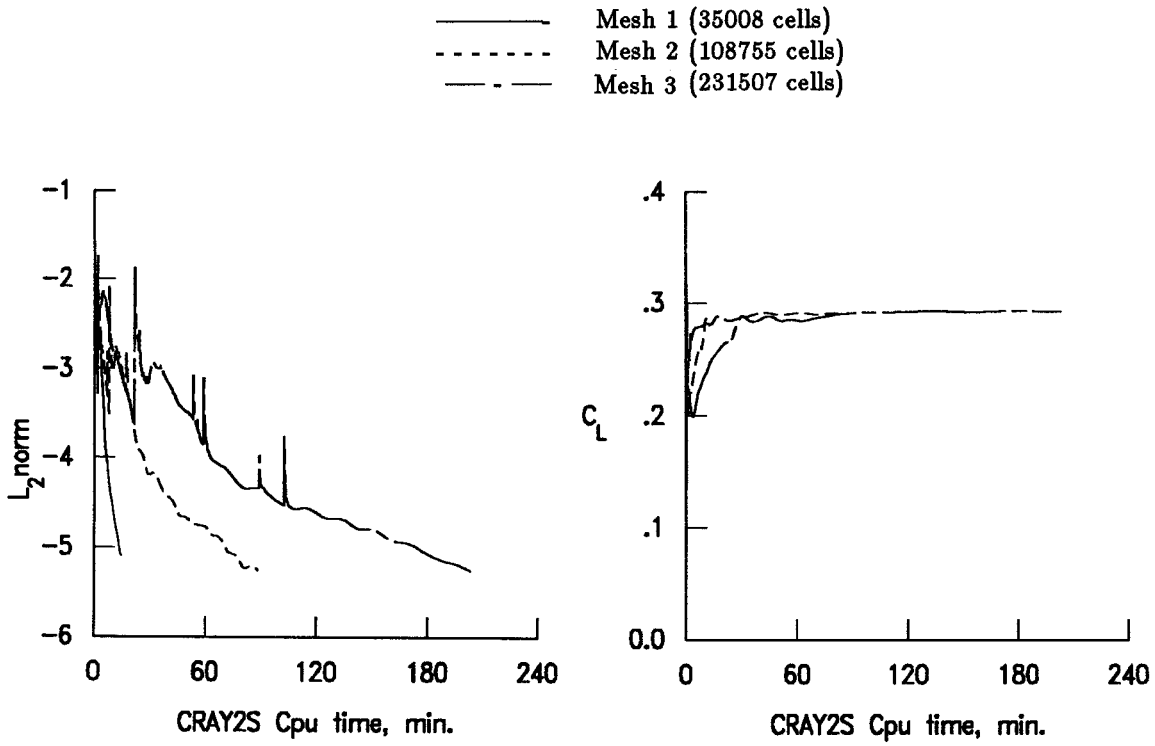


(a) Mesh 2.



(b) Mesh 3.

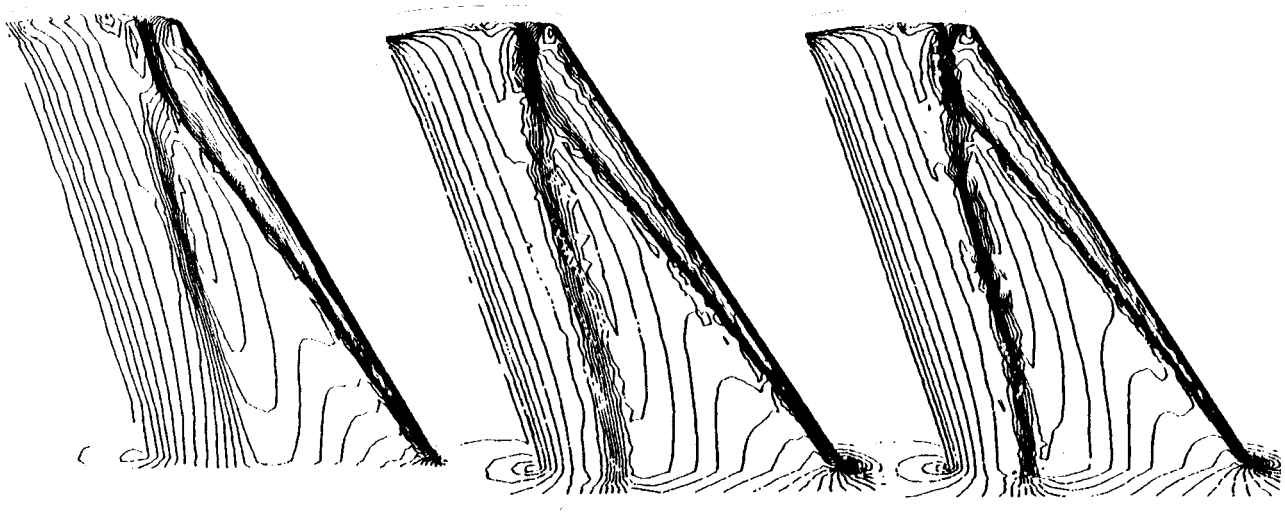
Figure 2.- Upper surface mesh for ONERA M6 wing.



(a) L^2-norm

(b) Lift coefficient

Figure 3.- Effect of mesh size on convergence history of ONERA M6 wing. $M_\infty = 0.84, \alpha = 3.06°$.



(a) Mesh 1

(b) Mesh 2

(c) Mesh 3

Figure 4.- Upper surface pressure contours for ONERA M6 wing. $M_\infty = 0.84, \alpha = 3.06°, \Delta(p/p_\infty) = 0.02$.

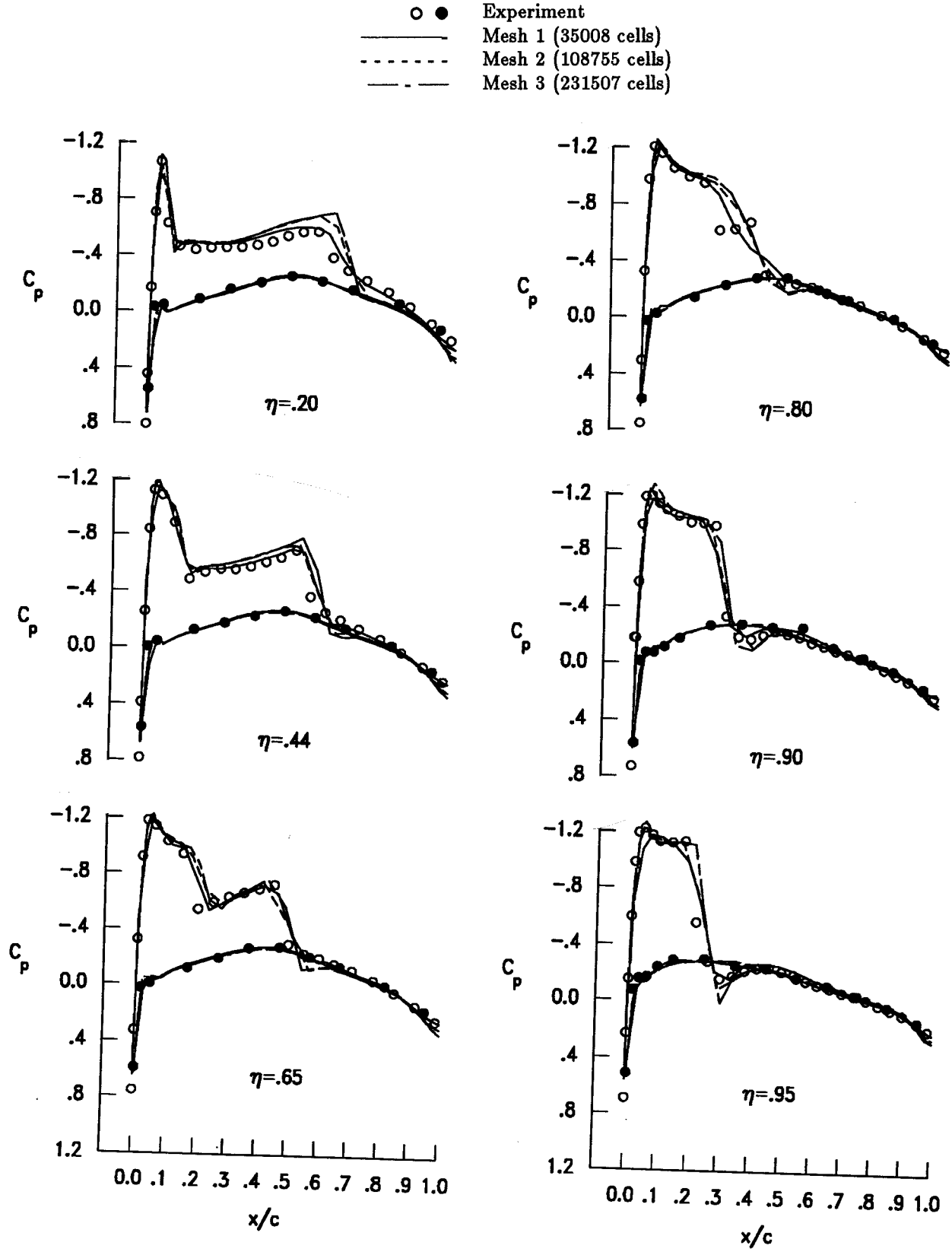
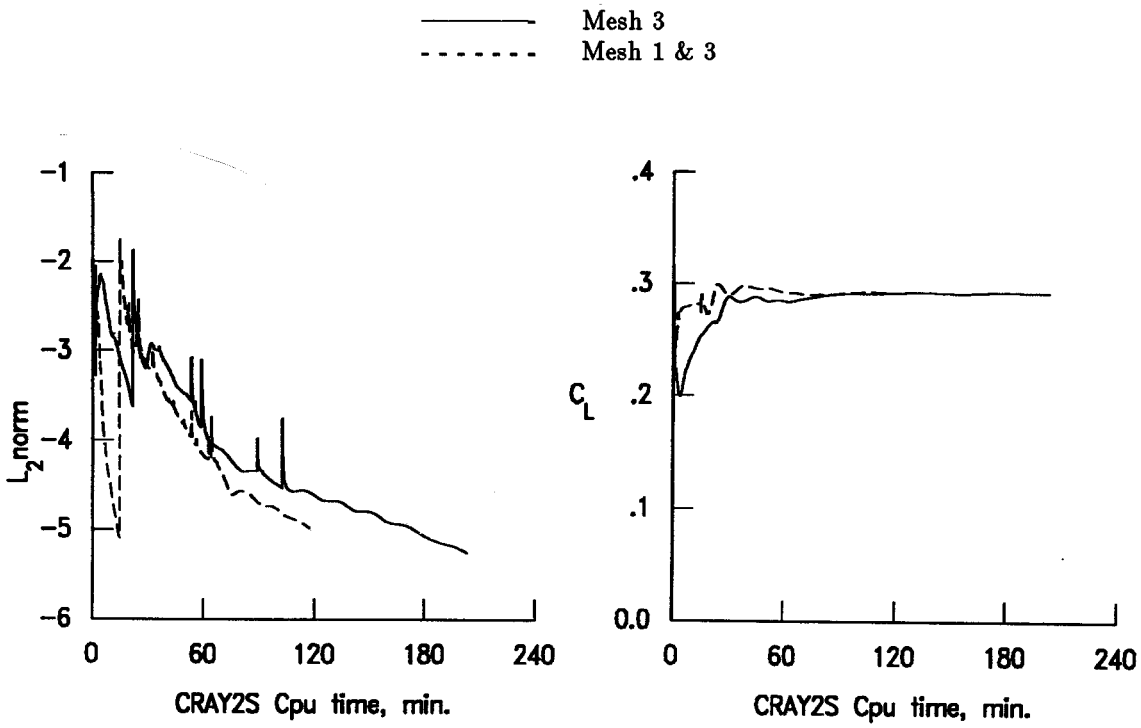


Figure 5.- Effect of mesh size on chordwise C_p distribution for ONERA M6 wing. $M_\infty = 0.84,\ \alpha = 3.06°$.



(a) L_2-norm

(b) Lift coefficient

Figure 6.- Effect of mesh sequencing on convergence history of ONERA M6 wing. $M_\infty = 0.84, \alpha = 3.06°$.

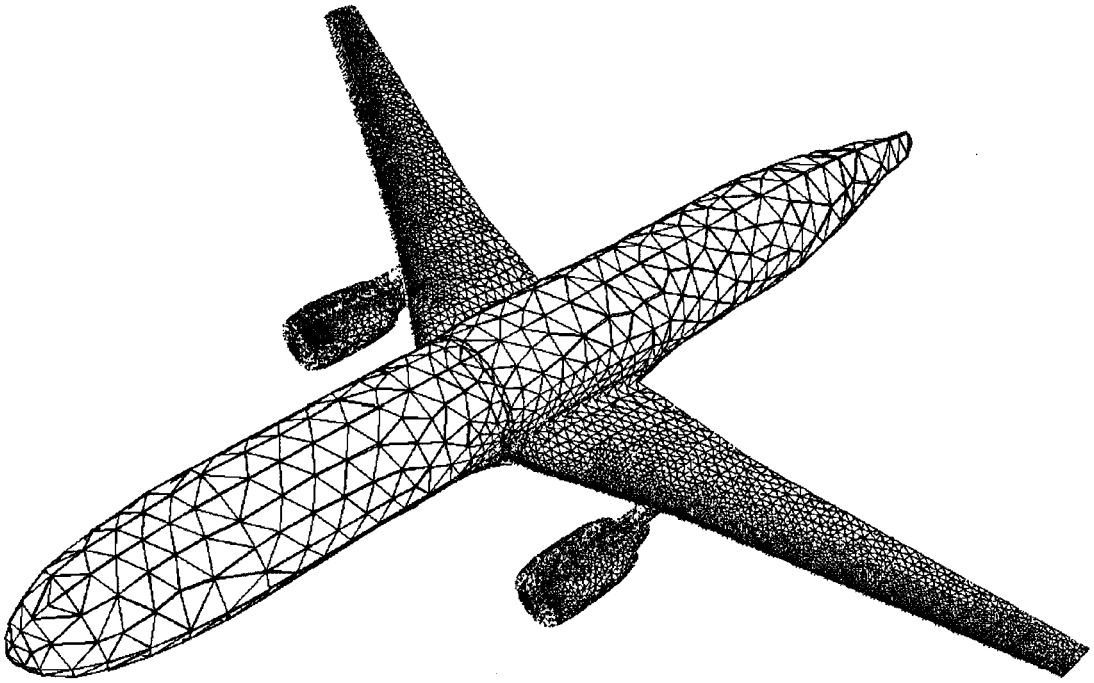


Figure 7.- Surface grid for Low-Wing Transport.

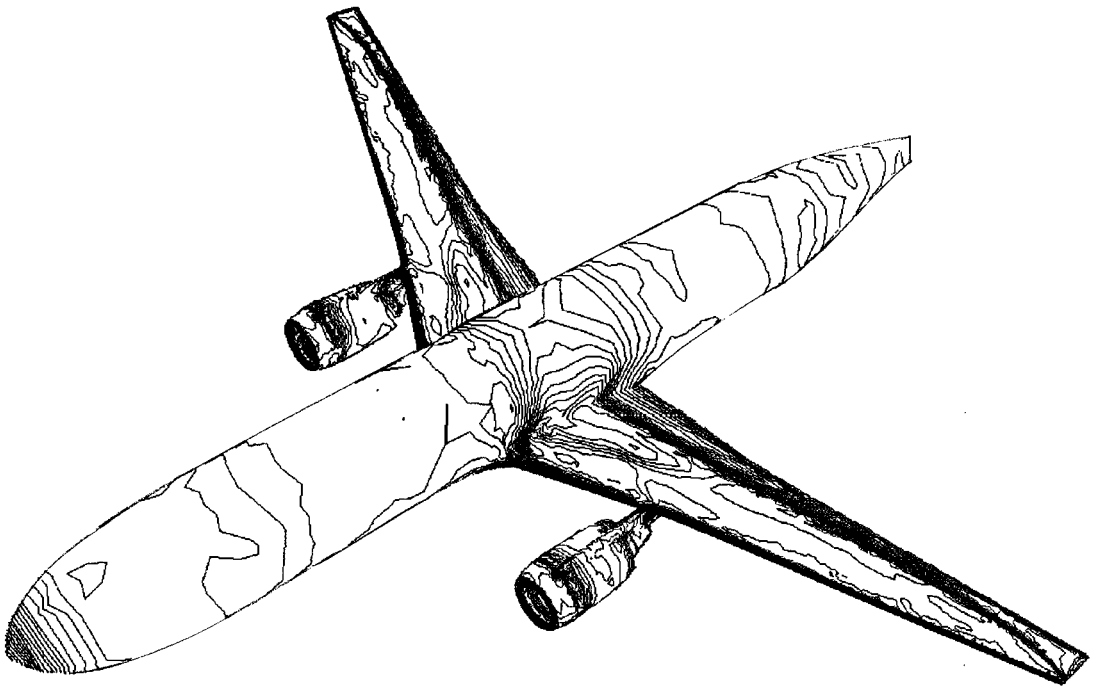


Figure 8.- Surface pressure contours for Low-Wing Transport.
$M_\infty = 0.768, \alpha = 1.116°$.

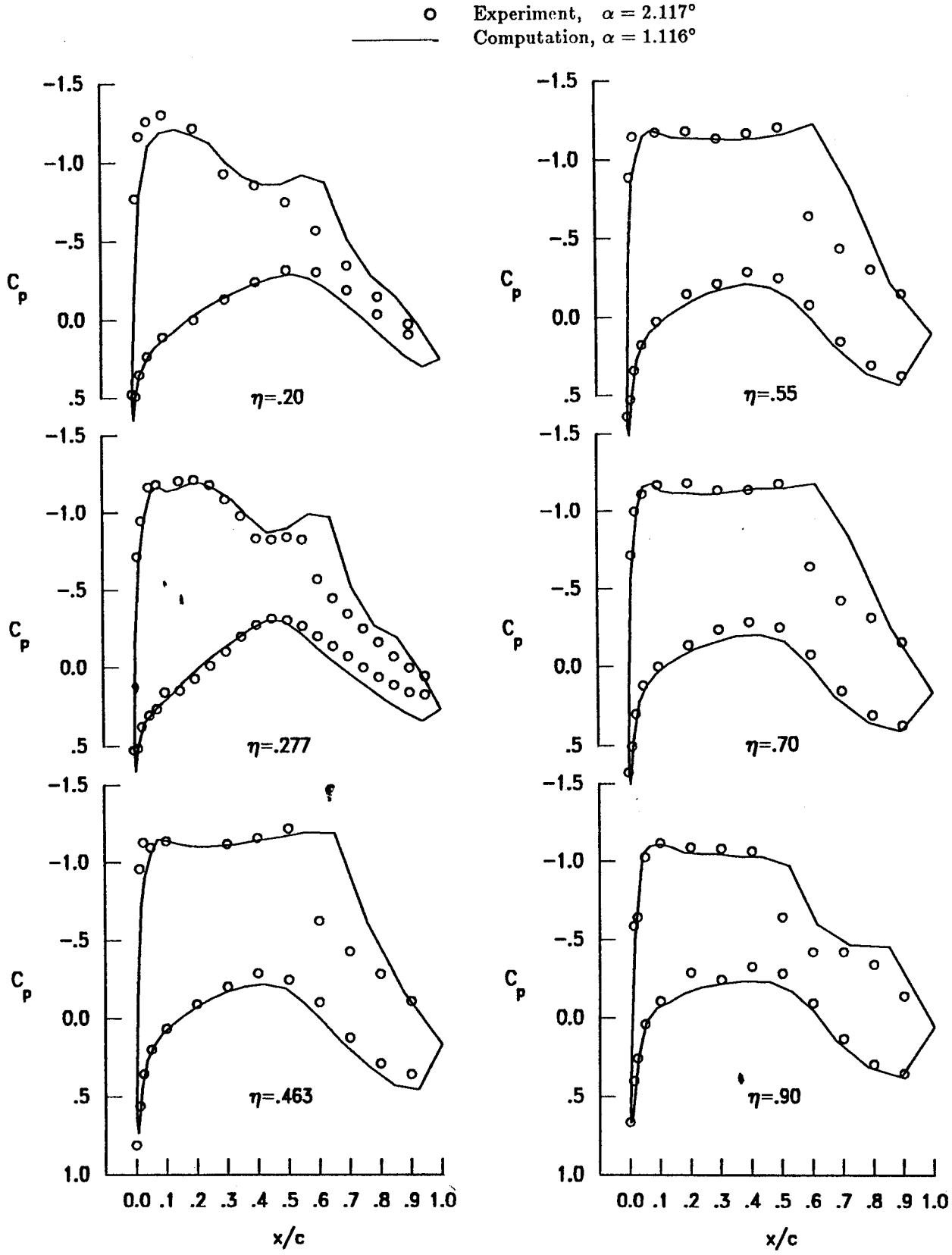


Figure 9.- Comparison of experimental data with unstructured inviscid solution. Low-Wing Transport, $M_\infty = 0.768$.

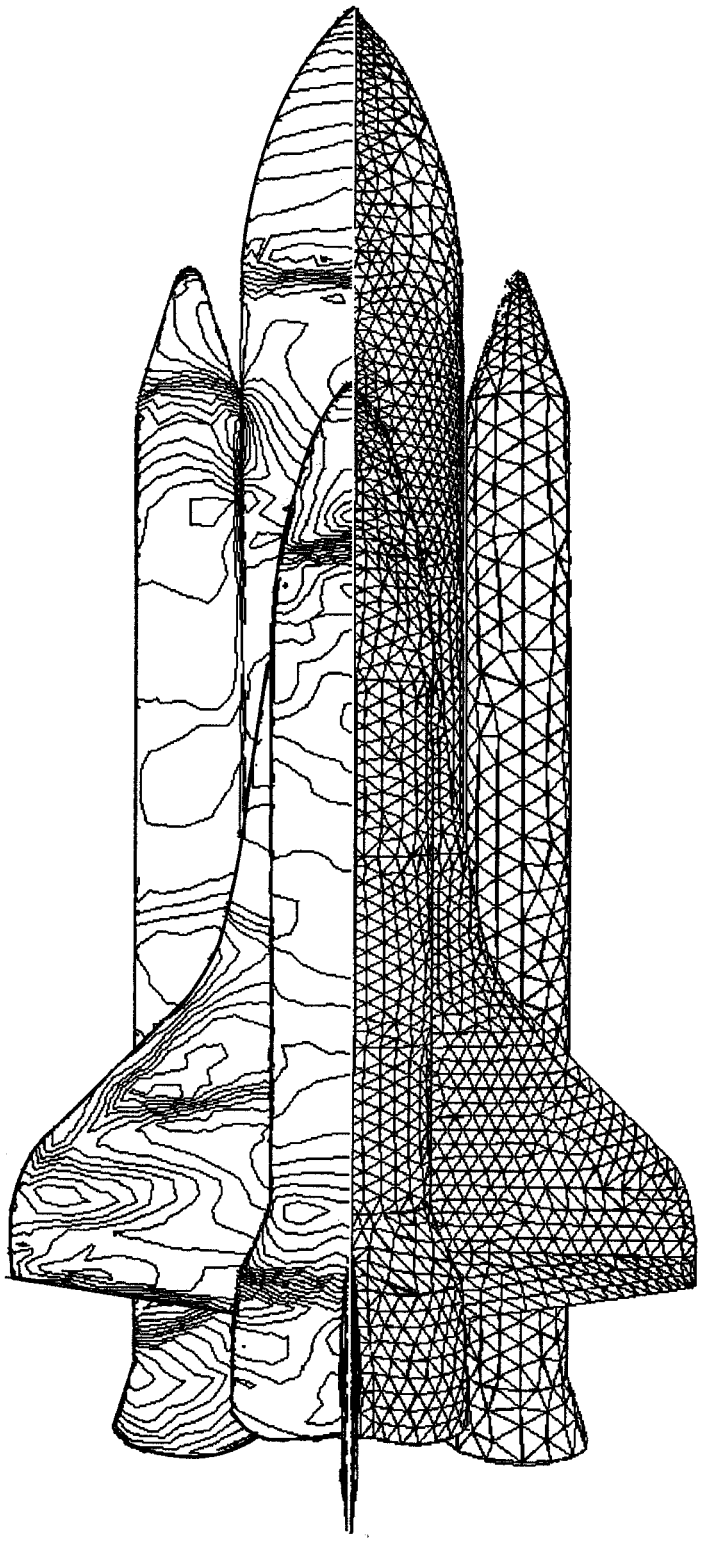


Figure 10.- Surface grid and pressure contours on STS.
$M_\infty = 1.05, \alpha = -3°$.

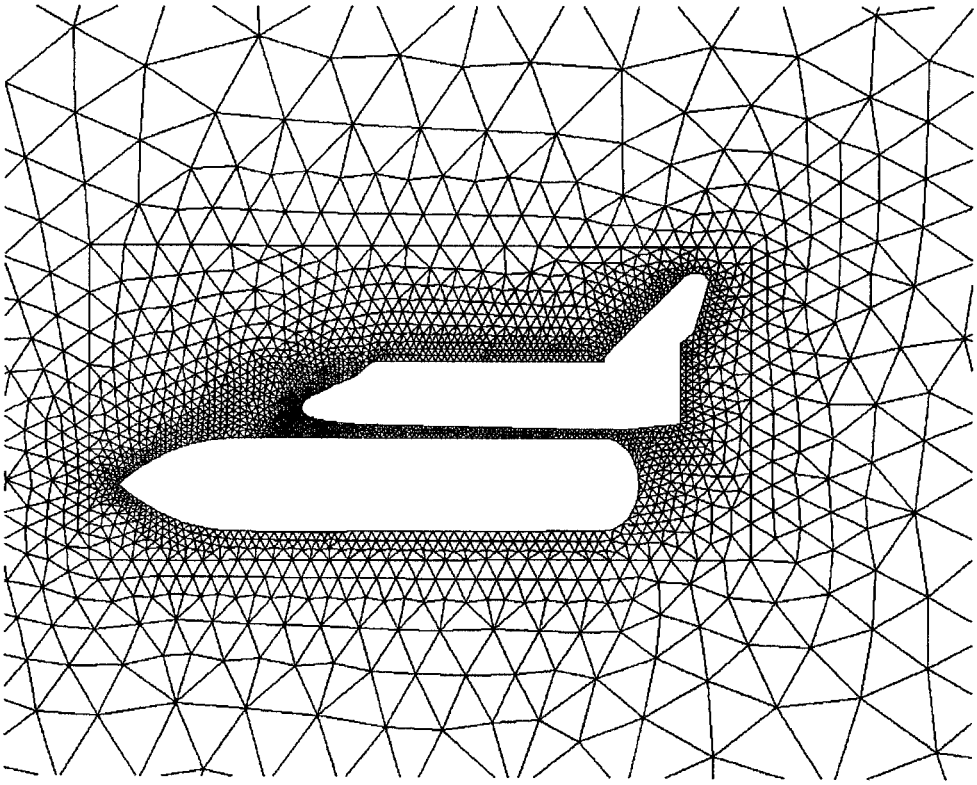


Figure 11.- Grid on symmetry plane for STS.

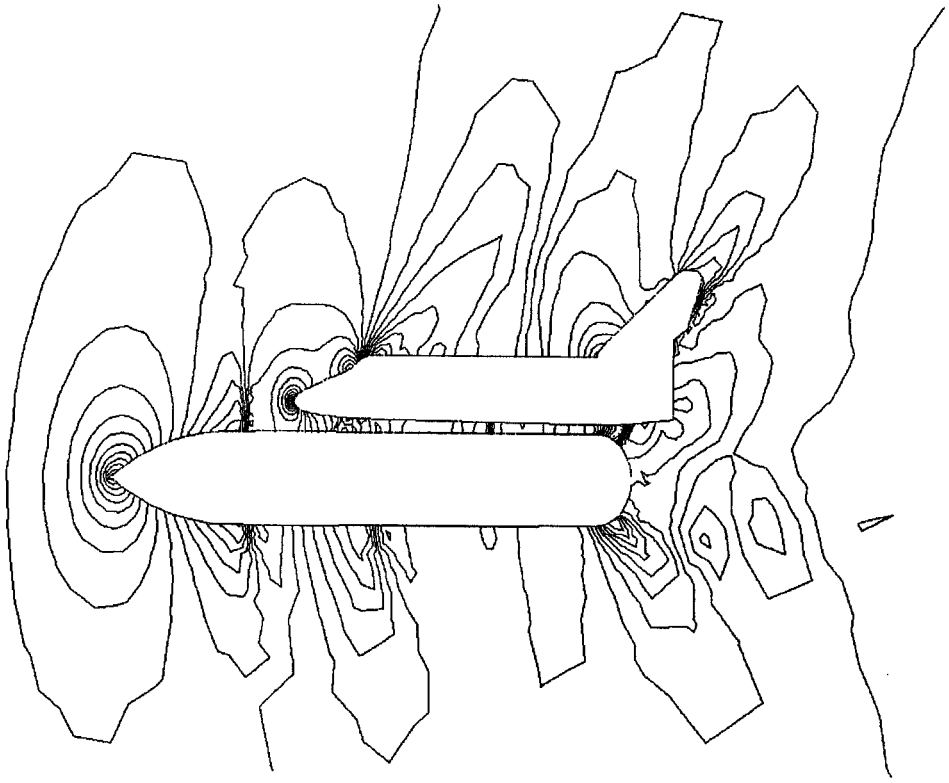


Figure 12.- Pressure contours on symmetry plane for STS.
$M_\infty = 1.05, \alpha = -3°$.