EULER CALCULATIONS FOR A COMPLETE AIRCRAFT

A. Jameson and T.J. Baker
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544
U.S.A.

Abstract

We describe a new finite element method for solving the Euler equations, and present the results of a transonic flow calculation for a commercial aircraft with pylon mounted engines. The finite element method uses a tetrahedral mesh, and establishes conservation of mass, momentum and energy in polyhedral control volumes by summing fluxes through the faces of the tetrahedra. The tetrahedra are generated by using a DeLaunay triangulation to connect a swarm of mesh points surrounding the aircraft.

Introduction

During the last two decades the science of aerodynamics has been transformed by the widespread introduction of computational methods to treat previously intractable problems, such as the calculation of transonic flows. Improvements in high speed electronic computers have made it feasible to attempt numerical calculations of progressively more complicated mathematical models of fluid flow, and to apply these methods to increasingly elaborate geometric configurations. Following the introduction of panel methods for subsonic flow in the sixties [1-2], and major advances in the simulation of transonic flow by the potential flow approximation in the seventies [3-6], the eighties have seen rapid developments in methods for solving the Euler and Navier Stokes equations [7-12].

A major pacing item of the emergence of a capability to treat a complete aircraft has been the development of a suitable method of mesh generation. For simple wing body combinations it is possible to generate rectilinear meshes without too much difficulty [9]: for more complicated configurations containing, for example, pylon mounted engines, it becomes increasingly difficult to produce a structured mesh which is aligned with all solid surfaces. An alternative is to use tetrahedral cells in an unstructured mesh which can be adapted to conform to the complex surface of an aircraft. Finite element methods of solving the potential flow and Euler equations on triangular and tetrahedral meshes have been developed by several authors [6,13,14]. Bristeau, Glowinski, Periaux, Perrier, Pironneau and Poirer achieved a striking success in solving the potential flow equation to predict the flow around a Falcon 50. Nevertheless, the generation of a tetrahedral mesh around configurations of such complexity remains a formidable problem. This paper describes a new finite element method for solving the Euler equations on a tetrahedral mesh, and its application to the calculation of transonic flow around a complete commercial aircraft with pylon mounted engines. Preliminary results were presented at the AIAA 24th Aerospace Sciences Meeting in January [15].

The finite element approximation is obtained by directly approximating the integral equations for conservation of mass, momentum and energy in polyhedral control volumes. The scheme can also be regarded as a *Galerkin method* in which the test function space is the set of piecewise linear tetrahedral elements: this can be shown to be equivalent to a flux balance based on polyhedral control volumes formed by the union of tetrahedra meeting at a common vertex. It turns out that each face is associated with precisely two such control volumes. It is therefore possible to reformulate the calculation in a particularly elegant way, in which the fluxes are evaluated in a single main loop over the faces. This novel decomposition leads to a substantial reduction in computational complexity. Steady state solutions are obtained by integrating the time dependent equations with a multistage time stepping scheme.

Convergence is accelerated by the use of locally varying time steps, residual averaging and enthalpy damping.

A new method is used to generate the tetrahedral mesh. Separate meshes are first generated around the individual aircraft components to create a cluster of points surrounding the whole aircraft. We do not require any regularity in this initial point distribution, only that a reasonable point density is created corresponding to the anticipated variation in the flowfield. The swarm of mesh points is then connected together to form tetrahedral cells which provide the basis for a single finite element approximation for the entire domain. This use of triangulation to unify separately generated meshes bypasses the need to devise interpolation procedures for transferring information between overlapping grids. The triangulation of a set of points to form disjoint tetrahedra is in general nonunique: our procedure is to generate the Delaunay triangulation [16-19]. This is dual to the Voronoi diagram that results from a division of the domain into polyhedral neighborhoods, each consisting of the subdomain of points nearer to a given mesh point that any other mesh point. The implementation of this method and the need to maintain the integrity of solid surfaces present a number of interesting problems. Although the Delaunay triangulation and associated Voronoi diagram has been exploited by others as a natural setting for calculations involving irregularly spaced points [16-17], we believe that the use of the Delaunay criterion as an explicit method of generating meshes for complex shapes is a new departure.

2. Finite Element Approximation

Let p, ρ, u, v, w, E and H denote the pressure, density, Cartesian velocity components, total energy and total enthalpy. For a perfect gas

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

where γ is the ratio of specific heats. The Euler equations for flow of a compressible inviscid fluid can be written in integral form as

$$\frac{\partial}{\partial t}\iiint_{\Omega} w\,d\Omega + \iint_{\partial\Omega} \underline{F}\cdot d\underline{S} = 0 \tag{1}$$

for a domain Ω with boundary $\partial\Omega$ and directed surface element $\underline{dS}$. Here w represents the conserved quantity and $\underline{F}$ is the corresponding flux. For mass conservation

$$w = \rho, \quad \underline{F} = (\rho u, \rho v, \rho w)$$

For momentum conservation

$$w = \rho u, \quad \underline{F} = (\rho u^2 + p, \quad \rho uv, \quad \rho uw)$$

with y and z momentum quantities similarly defined, and for energy conservation

$$w = \rho E, \quad \underline{F} = (\rho Hu, \quad \rho Hv, \quad \rho Hw)$$

Consider the differential form of equation (1)

$$\frac{\partial w}{\partial t} + \nabla\cdot\underline{F} = 0$$

Multiplying by a test function ϕ and integrating by parts over space leads to

$$\frac{\partial}{\partial t}\iiint_{\Omega} \phi w\,d\Omega = \iiint_{\Omega} \underline{F}\cdot\nabla\phi\,d\Omega - \iint_{\partial\Omega} \phi\underline{F}\cdot d\underline{S} \tag{2}$$

Suppose now that we take ϕ to be the piecewise linear function with the value unity at one node (denoted by 0 in Figure 1), and zero at all other nodes. Then the last term vanishes except in the case when 0 is adjacent to the boundary. Also $\nabla\phi$ is constant in every tetrahedron, and differs from zero only in the tetrahedra with a common vertex at node 0. Since ϕ_x is constant in a tetrahedron it may be evaluated as

$$\phi_x = \frac{1}{V}\iiint \phi_x \; dxdydz = \frac{1}{V}\sum_k S_{x_k}\bar{\phi}_k$$

where V is the cell volume, S_{x_k} and $\bar{\phi}_k$ are projected area of the kth face in the x

direction and the average value of ϕ on the kth face, and the sum is taken over the faces of the tetrahedron. For the given test function $\bar{\phi} = 1/3$ on the faces 012, 023, and 031 and zero on the face 123. Also the projected area S_x on face 123 is equal and opposite to the sum of the projected face areas of the other three faces. Using the same procedure to evaluate ϕ_y and ϕ_z, it follows that

$$\nabla\phi = -\underline{S}/3V \tag{3}$$

where $\underline{S}$ is the directed area of the face opposite vertex 0. Now treat $\underline{F}$ as piecewise linear and use equation (3) to evaluate the volume integral on the right side of equation (2). Then each tetrahedron meeting at node 0 introduces a contribution $(\bar{\underline{F}} \cdot \underline{S})/3$ where $\bar{\underline{F}}$ is the average value of $\underline{F}$ in the cell. For the cell illustrated in Figure 1, for example,

$$\bar{\underline{F}} = \frac{1}{4}(\underline{F}_0 + \underline{F}_1 + \underline{F}_2 + \underline{F}_3)$$

Summing over all cells meeting at node 0 leads to the total contribution

$$\frac{1}{3}\sum_k \underline{F}_k \cdot \underline{S}_k$$

Since the control volume is closed, however,

$$\sum_k \underline{S}_k = 0$$

Therefore the contribution of F_0 to $\bar{F}_k$ can be discarded, leading to a sum over the faces multiplied by a constant. Thus if we write

$$\tilde{\underline{F}} = \frac{1}{3}(\underline{F}_1 + \underline{F}_2 + \underline{F}_3)$$

for the average value of $\underline{F}$ on the face opposite vertex 0 we find that the right-hand side of equation (2) can be replaced by

$$-\frac{1}{4}\sum \tilde{\underline{F}}_k \cdot \underline{S}_k$$

On the left-hand side of equation (2) we take w to be constant inside the control volume. Since ϕ is piecewise linear, the volume average value is $\bar{\phi} = 1/4$. The factor 1/4 cancels on each side and the approximation to equation (2) can therefore be written as

$$\frac{d}{dt}\left(\sum_k V_k\right) w + \sum \tilde{\underline{F}}_k \cdot \underline{S}_k = 0 \tag{4}$$

Referring to Figure 2, which illustrates a two dimensional mesh, it may be seen that with a triangular or tetrahedral mesh, each face is a common external boundary to exactly two control volumes. Therefore each internal face can be associated with a set of 5 mesh points consisting of its three corners 1, 2 and 3, and the vertices 4 and 5 of the two tetrahedra based on the face, as illustrated in Figure 3. Vertices 4 and 5 are the centers of the two control volumes influenced by the face. It is now possible to generate the approximation (4) by presetting the flux balance at each mesh point to zero, and then performing a single loop over the faces. For each face one first calculates the fluxes of mass, momentum and energy across the face, and then one assigns these contributions to the vertices 4 and 5 with positive and negative signs respectively. Since every contribution is transferred from one control volume into another, all quantities are perfectly conserved. Mesh points on the inner and outer boundaries lie on the surface of their own control volumes, and the accumulation of the flux balance in these volumes has to be correspondingly modified. At a solid surface it is also necessary to enforce the boundary condition that there is no convective flux through the faces contained in the surface.

3. Dissipation

Equation (4) represents a nondissipative approximation to the Euler equations. Dissipative terms may be needed for two reasons; to eliminate the occurrence of undamped or lightly damped nodes, and to prevent oscillations near shock waves.

The simplest form of dissipation is to add a term generated from the difference between the value at a given node and its nearest neighbors. That is, at node 0, we add a term

$$D_0 = \sum_k \varepsilon^{(1)}_{k0} (w_k - w_0) \tag{5}$$

where the sum is over the nearest neighbors, as illustrated in Figure 4. The contribution $\varepsilon^{(1)}_{k0}(w_k - w_0)$ is balanced by a corresponding contribution $\varepsilon^{(1)}_{k0}(w_0 - w_k)$ at node k, with the result that the scheme remains conservative. The coefficients $\varepsilon^{(1)}_{k0}$ may incorporate metric information depending on local cell volumes and face areas, and can also be adapted to gradients of the solution. It is shown in reference 15 that the addition of properly controlled differences along edges can be used to assure a positivity condition on the coefficients of the semi-discrete scheme, which will prevent growth in the maximum norm and inhibit oscillations is the solution.

Formula (5) is no better than first order accurate unless the coefficients are proportional to the mesh spacing. A more accurate scheme is obtained by recycling the edge differencing procedure. After first setting

$$E_0 = \sum_k (w_k - w_0) \tag{6}$$

at every mesh point, one then sets

$$D_0 = - \sum_k \varepsilon^{(2)}_{0k} (E_k - E_0) \tag{7}$$

An effective scheme is produced by blending formulas (5) and (7), and adapting $\varepsilon^{(1)}_{0k}$ to the local pressure gradient. This is accomplished by calculating

$$P_0 = \sum_k \left| \frac{P_k - P_0}{P_k + P_0} \right|$$

at every mesh point, and then taking $\varepsilon^{(1)}_{0k}$ proportional to max (P_0, P_k). Formulas of this type have been found to have good shock capturing properties, and the required sums can be efficiently assembled by loops over the edges.

4. Integration to a Steady State

The discretization procedures of Sections 2 and 3 leads to a set of coupled ordinary differential equations, which can be written in the form

$$\frac{dw}{dt} + R(w) = 0 \tag{8}$$

where w is the vector of the flow variables at the mesh points, and R(w) is the vector of the residuals, consisting of the flux balances defined by equation (4), together with the added dissipative terms. These are to be integrated until they reach a steady state.

For this purpose we use a multistage time stepping scheme of the same type which has proved effective in calculations on rectilinear meshes. Let w^n be the result after n steps. To advance one step Δt with an m stage scheme we set

$$\begin{aligned} w^{(0)} &= w^n \\ w^{(1)} &= w^{(0)} - \alpha_1 \Delta t R^{(0)} \\ &\cdots \\ w^{(m-1)} &= w^{(0)} - \alpha_{m-1} \Delta t R^{(m-2)} \\ w^{(m)} &= w^{(0)} - \Delta t R^{(m-1)} \\ w^{n+1} &= w^{(m)} \end{aligned}$$

The residual in the (q+1)-st stage is evaluated as

$$R^{(q)} = \frac{1}{V} \sum_{r=0}^{q} \{ \beta_{qr} Q(w^{(r)}) - \gamma_{qr} D(w^{(r)}) \}$$

where $Q(w)$ is the approximation to the Euler equations and $D(w)$ represents the dissipative terms, and the coefficients β_{qr} and γ_{qr} satisfy the consistency condition that

$$\sum_{r=0}^{q} \beta_{qr} = \sum_{r=0}^{q} \gamma_{qr} = 1$$

In practice a three stage scheme in which the dissipative terms are evaluated only once has proved effective. For this scheme

$$\alpha_1 = .6 \ , \quad \alpha_2 = .6$$
$$\beta_{qq} = 1 \ , \quad \beta_{qr} = 0 \ , \ q > r$$
$$\gamma_{q0} = 1 \ , \quad \gamma_{qr} = 0 \ , \ r > 0$$

Convergence to a steady state is accelerated by using a variable time step close to the stability limit at each mesh point. The scheme is accelerated further by the introduction of residual averaging [9]. At the mesh point 0 the residual R_0 is replaced by $\tilde{R}_0$ where $\tilde{R}_0$ is an approximation to the solution $\bar{R}_0$ of the equation

$$\bar{R}_0 + \sum_k \varepsilon(\bar{R}_0 - \bar{R}_k) = R_0 \tag{9}$$

in which the sum is over the nearest neighbors. This is similar to the weighted average appearing in the Galerkin method, but with the opposite sign for the coefficient ε, leading to an increase in the permissible time step instead of a reduction. In practice it has been found effective to obtain $\bar{R}$ by using two steps of the Jacobi iteration

$$\tilde{R}_0^{(m)} + \sum_k \varepsilon(\tilde{R}_0^{(m)} - \tilde{R}_k^{(m-1)}) = R_0 \tag{10}$$

starting from $\tilde{R}_0^{(0)} = R_0$.

5. Mesh Generation

The triangulation procedure will connect an arbitrary collection of points to form a tetrahedral mesh. If the aircraft surface is adequately defined, we can introduce the aircraft into some pre-defined cloud of points, remove all points lying inside the aircraft structure, and then connect up the remaining points including a prescribed set of points lying on the aircraft surface. In the present version of our code we have chosen to make use of existing mesh generation techniques to create a cloud of points around the wing/body/tail/fin combination, and a further cloud of points around each nacelle.

A mesh for the wing/body/tail combination is generated by the procedure used in FL059 [9,20]. This starts with a C-mesh around the wing which is generated by the introduction of sheared parabolic coordinates. This is accomplished in two stages. First we define a parabolic mapping which unwraps the wing to a shallow bump above a half plane. Let $\underline{X} = (X,Y,Z)$ be a point in the mapped space corresponding to $\underline{x}$ in the physical space. The unwrapping transformation is

$$\underline{X} = P_w \, \underline{x}$$

where P_w is defined by

$$x - x_0(z) = X^2 - Y^2$$
$$y - y_0(z) = 2XY$$
$$z = Z$$

and $\underline{x}_0$ is a point just inside the wing leading edge. The bump is then removed by a shearing transformation. Let $Y_w(X,Z)$ be the surface of the wing in mapped space. We define a shearing S_w taking $X' = (X', Y', Z')$ to $\underline{X}$ by the transformation

$$\underline{X} = S_w \, \underline{X'}$$

where S_w is defined by

$$X = X'$$

$$Y = Y' + Y_w(X,Z)$$
$$Z = Z'$$

This maps the half space $Y' \geq 0$ onto the region in $\underline{X}$ - space above Y_w. The C mesh is then generated by introducing rectangular coordinates in the half-space, and reversing the transformations.

The mesh around a combination of a wing plus body is generated by introducing a further transformation which maps an arbitrary shaped body into the symmetry plane, $z = 0$. This mapping can be constructed as a combination of a Joukowski mapping plus a shearing. This sequence of operations will generate a mesh that conforms with the body surface but such that the crest line of the body is not necessarily aligned with any mesh line. This deficiency is rectified by deforming the mesh lines in the mapped space to ensure that the resulting mesh is completely aligned with the body surface. The extension of these ideas to include a tail and fin follows the same principle of first utilizing a mapping to simplify the configuration, fitting a mesh in the mapped space and then mapping back to obtain the mesh in physical space.

Separate meshes are generated for each nacelle, again using a combination of unwrapping plus shearing. In this case we define a mapping P_N by the conformal transformation

$$\xi = z - e^{-z}$$

where $z = x + iy$ and $\xi = X + iY$. Here x is a coordinate aligned with the nacelle axis, and y is the radial coordinate corresponding to a cylindrical coordinate system such that $y = 0$ is the nacelle axis. If the nacelle is not axisymmetric we take the axis of the cylindrical coordinate system to be an approximate center line through the nacelle. The z coordinates are scaled so that $y = \pi$ corresponds to a cut inside the nacelle section with the point $(0, i\pi)$ just inside the section leading edge. The mapping defined above is applied to each nacelle section, transforming the space around the nacelle onto the space inside a deformed cylinder with the nacelle surface mapped to the cylinder surface. A shearing transformation can now be combined with the inverse of the mapping to generate a mesh that is aligned with the nacelle surface. A straightforward extension of the sequential mapping procedure can be used to accommodate a center body. Finally we can generate points around the pylons by treating each pylon as an isolated wing and using the mapping sequence that has previously been described.

6. Delaunay Triangulation

If the set of points is denoted by $\{ P_i \}$, the Voronoi neighborhood of the point P_i is defined as the region of space

$$V_i = \{ x \mid d(x,P_i) < d(x, P_j) \text{ for all } i \neq j \}$$

Here x is a point in three dimensional Euclidean space and d is the Euclidean metric. Each such region V_i is the intersection of the open half spaces bounded by the perpendicular bisectors of the lines joining P_i to each of the other P_j. The regions are thus convex polyhedra and, in general, four such regions meet at each vertex of the Voronoi diagram. We refer to regions that have common boundary faces as contiguous and likewise denote the points associated with two such regions as contiguous points. For each vertex of the Voronoi diagram we can join the four contiguous points, which have that vertex in common, by four planes to form a tetrahedron. The aggregate of tetrahedra forms the unique triangulation of the convex hull of points $\{ P_i \}$ known as the Delaunay triangulation. Each Voronoi vertex is the circumcenter of the tetrahedron with which it is associated, and no point lies within the sphere that circumscribes the tetrahedron. This property ensures that the aspect ratio of the tetrahedra is reasonable and, in some sense, leads to an optimum triangulation for a given distribution of points.

The computation of the Voronoi diagaram and its associated triangulation has received considerable attention recently [18-19]. The algorithm used here is based on Boywer's method [19]. As Boywer notes, it is possible to record the structure of the triangulation by constructing two lists for each vertex in the structure. Each list has four entries: the first contains the forming points of the tetrahedron

associated with the vertex and the second list holds the addresses of the neighboring vertices. The process is sequential: each new point is introduced into the existing structure which is broken and then reconnected to form a new Delaunay triangulation. When a new point is introduced into the existing triangulation, it is first necessary to identify a vertex of the Voronoi diagram that will be deleted by the new point. As the vertex at the circumcenter of the tetrahedron in which the point lies must necessarily be deleted, we are assured that at least one deleted vertex can be identified. Next we look at the neighbors of the deleted vertex for other vertices of the Voronoi diagram that may be deleted. We continue the tree search, creating a list of deleted vertices until all deleted vertices have been identified. From the list of deleted Voronoi vertices, we can determine the neighboring contiguous vertices in the undeleted set. Each point lying on the interface with the deleted region is joined to the new point. The deleted region is necessarily simply connected and star shaped. The new tetrahedra thus formed will exactly fill the deleted region and, moreover, will also satisfy the Delaunay criterion. It remains to label the new Voronoi neighborhoods and revise the lists that record the data structure.

Our strategy is to triangulate the entire space including the interior of the aircraft as well as the exterior. It is then important to identify interior tetrahedra correctly, as these must be removed before carrying out the flow calculation. Furthermore, it is necessary to prevent connections from exterior points breaking through the aircraft surface. We start the triangulation by introducing the outer boundary and then the aircraft surface points, component by component. After all the surface points have been introduced the interior tetrahedra are identified. Subsequently, if the insertion of a new point would cause a reconnection penetrating the surface, that point is rejected from the triangulation. This will occur if the point lies inside the DeLaunay sphere of an interior tetrahedron. To allow the introduction of points close to the surface it is therefore essential to make sure that the DeLaunay spheres of all the interior tetrahedra are sufficiently small. After the initial triangulation of the surface points we check the size of the DeLaunay spheres. Then, if any of these exceed a predetermined threshold, we introduce additional surface points until no excessively large spheres remain before proceeding to the introduction of the flow field points.

7. Results

In Figure 5 we show the result of a transonic flow calculation for a Boeing 747-200 flying at Mach .84 and an angle of attack of 2.73 degrees. The result is displayed by computed pressured contours on the surface of the aircraft. Flow is allowed through the engine nacelles which are modelled as open tubes. The mesh contains 24685 points and 132793 tetrahedra. The calculation was performed at Cray Research on a Cray XMP 216: the complete calculation took 3924 seconds. Of these 1448 seconds were spent in generating the mesh points and triangulating them. The remaining 2476 seconds were spent in the flow computation, which was performed with 400 cycles of the three stage scheme. Implicit smoothing with a smoothing parameter $\varepsilon = 1$ allowed the use of time steps corresponding to a nominal Courant number of 5. The number of supersonic points was frozen after 200 cycles, and the average residual was reduced from $.335\times10^{2}$ to $.161\times10^{-3}$ after 400 cycles. Although the mesh is fairly coarse, the significant features of the flow are evident, including the interference effects of the wing and tail on the body, and the mutual interference of the wing, nacelle and pylon.

Calculations with this number of mesh points require slightly more than 8 million words of memory. Within the limit of 16 million words available on a Cray XMP 216 it should be possible to introduce nearly twice as many mesh points to produce a mesh with about 1/4 million tetrahedrons. This should be sufficient to resolve the main features of the flow over the complete configuration. Eventually, in order to provide a detailed representation of the aircraft, we anticipate the need to increase the number of mesh points by a factor of between five and ten. This will require access to machines with a much larger memory, such as the Cray 2.

8. Conclusion

The results for the Boeing 747 clearly establish the feasability of our approach. We are now pursing the development of a variety of improvements and extensions of the

method. These include:

(1) Vectorization

Vectorization of the main loops has already been achieved by separating the cells, faces and edges into groups such that no vertex at which contributions are being accumulated is referred to more than once in each group. Using this procedure, rates of computation ranging from 17-38 megaflops have been realized on a Cray XMP computer, depending on the mesh. These variations stem from variations in the sizes of the groups and the associated vector lengths. The efficiency can be improved by making sure that no group is too small. The analysis of the associated sorting problems leads to some general map coloring problems: for example, what is the minimum number of colors needed to color the tetrahedra in such a way that tetrahedra meeting at the same vertex do not have the same color.

(2) Improved Distribution of Mesh Points

The DeLaunay triangulation procedure connects an arbitrary cluster of points to form a tetrahedral mesh. It can be anticipated, however, that the accuracy will be improved by ensuring a favorable distribution of the points, with sufficient concentration in the neighborhood of the surface, and particularly in critical regions such as the pylon wing intersection. The present mesh generating procedure needs to be improved to provide better control of the size and aspect ratio of the tetrahedra.

(3) Adaptive Mesh Refinement

The unstructured tetrahedral mesh provides a natural setting for the introduction of an adaptive mesh refinement procedure in which additional mesh points are inserted in regions where there are rapid variations in the flow, or an indication of relatively large discretization error. This provides a method of reducing the thickness, for example, of a computed shock layer. The promise of this approach has already been demonstrated in the work of Löhner, Morgan and Peraire [21], and Holmes and Lamson [22].

(4) Multigrid Acceleration

It should be possible to make a further reduction in the cost of the flow calculation by using multiple grids to accelerate the convergence to a steady state. Since the meshes are unstructured, no simple relationship can be assumed between a coarse and a fine mesh, and rather complex procedures must be used to transfer data between the meshes.

(5) Extension to Navier Stokes Equations

By using the weak form, equation (2), the viscous terms of the Navier Stokes equations can rather easily be approximated within the present framework. Then, as a result of the integration by parts, only first derivatives of the velocities are needed to evaluate the rate of strain and stress tensors. These may be taken as constant in each tetrahedron, consistent with the assumption of linear variation in each element. A new version of the program containing additional subroutines to evaluate the viscous terms is currently under development.

(6) Simulation of Engine Power Effects

The present model allows free flow through the engine nacelles. A more realistic simulation can be achieved by introducing source terms to represent the engine power effects.

Acknowledgments

Most of our work on the airplane computer program has been carried out on computers belonging to the Cray Research Corporation. We should like to thank Cray Research for providing us with access to their computers and we are particularly grateful to Kent Misegades for his help and support. Substantial financial support for our work has been provided by the IBM Corporation, the Office of Naval Research, and the NASA Langley Research Center.

References

1. Hess, J. L. and Smith, A. M. O., "Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies", Douglas Aircraft Report, ES 40622, 1962.

2. Rubbert, P. E. and Saaris, G. R., "A General Three Dimensional Potential Flow Method Applied to V/STOL Aerodynamics", SAE Paper 680304, 1968.

3. Murman, E.M. and Cole, J.D., "Calculation of Plane Steady Transonic Flows", AIAA Journal, Vol. 9, 1971, pp. 114-121.

4. Jameson, Antony, "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1", Comm. Pure. Appl. Math, Vol. 27, 1974, pp. 283-309.

5. Jameson, Antony and Caughey, D. A., "A Finite Volume Method for Transonic Potential Flow Calculations", Proc. AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, 1977, pp. 35-54.

6. Bristeau, M. O., Pironneau, O., Glowinski, R., Periaux, J., Perrier, P., and Poirier, G., "On the Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least Squares and Finite Element Methods (II). Application to Transonic Flow Simulations", Proc. 3rd International Conference on Finite Elements in Nonlinear Mechanics, FENOMECH 84, Stuttgart, 1984, edited by J. St. Doltsinis, North Holland, 1985, pp. 363-394.

7. 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, AIAA 14th Fluid Dynamics and Plasma Dynamics Conference, Palo, Alto, 1981.

8. Ni R.H., "A Multiple Grid Scheme for Solving the Euler Equations", Proc. AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, 1981, pp. 257-264.

9. Jameson, Antony, and Baker, Timothy J., "Solution of the Euler Equations for Complex Configurations", Proc. AIAA 6th Computational Fluid Dynamics Conference, Danvers, 1983, pp. 293-302.

10. Jameson, Antony, "Multigrid Algorithms for Compressible Flow Calculations", Second European Conference on Multigrid Methods, Cologne, October 1985, Princeton University Report MAE 1743.

11. Pulliam, T.H.,and Steger, J.L., "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms", AIAA Paper 85-0360, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1985.

12. MacCormack, R.W., "Current Status of Numerical Solutions of the Navier-Stokes Equations", AIAA Paper 85-0032, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1985.

13. Lohner, R., Morgan, K., Peraire, J. and Zienkiewicz, O.C., "Finite Element Methods for High Speed Flows", AIAA Paper 85-1531, AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, July 1985.

14. Jameson, A., and Mavriplis, D., "Finite Volume Solution of the Two Dimesional Euler Equations on a Regular Triangular Mesh, AIAA Paper 83-0435, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1985.

15. Jameson, A., Baker, T.J. and Weatherill, N.P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft", AIAA Paper 86-0103, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January, 1986.

16. Augenbaum, J.M., "A Langrangian Method for the Shallow Water Equations Based on a Voronoi Mesh-One Dimensional Results", J. Comp. Physics, Vol. 53, No. 2, February 1984.

17. McCartin, B., "Discretization of the Semiconductor Device Equations", in

New Problems and New Solutions for Device and Process Modeling, pp.72-82, Bode Press, 1985.

18. Watson, D.F., "Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes", The Computer Journal, Vol. 24, No. 2, pp. 162-166.

19. Bowyer, A., "Computing Dirichlet Tessellations", The Computer Journal, Vol. 24, No. 2, pp. 162-166.

20. Baker, T.J., "Mesh Generation by a Sequence of Transformations", to appear in Applied Numerical Mathematics, December, 1986, Princeton University Report MAE 1739.

21. Lohner, R., Morgan, K., and Peraire, J., "Improved Adaptive Refinement Strategies for Finite Element Aerodynamic Configurations", AIAA Paper 86-0499, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

22. Holmes, D.G., and Lamson, S.H., "Adaptive Triangular Meshes for Compressible Flow Solutions", First International Conference on Numerical Grid Generation in Computational Dynamics, Landshut, W. Germany, July 1986.

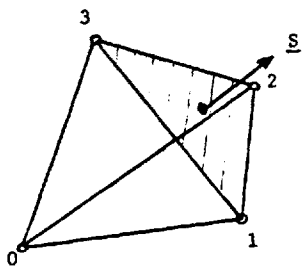

Figure 1. One tetrahedron of the control volume centered at node 0.

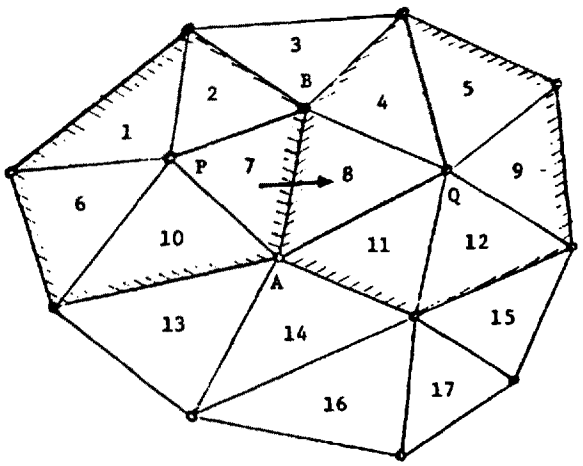

Figure 2. A triangular mesh in 2 dimensions: The control volume at P is the union of triangles 1, 6, 10, 7 and 2, while that at Q is the union of triangles 4, 8, 11, 12, and 9. The flux across the edge AB is from the control volume at P to the control volume at Q.

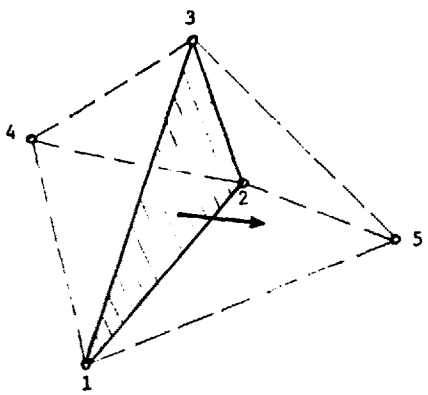

Figure 3. Flux through face defined by nodes 1, 2 and 3 is out of the control volume centered at node 4 and into the control volume centered at node 5.

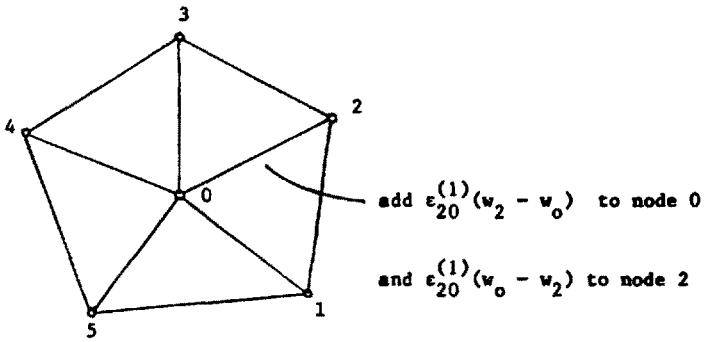

Figure 4. Construction of dissipation from differences along edges in a two dimensional mesh.

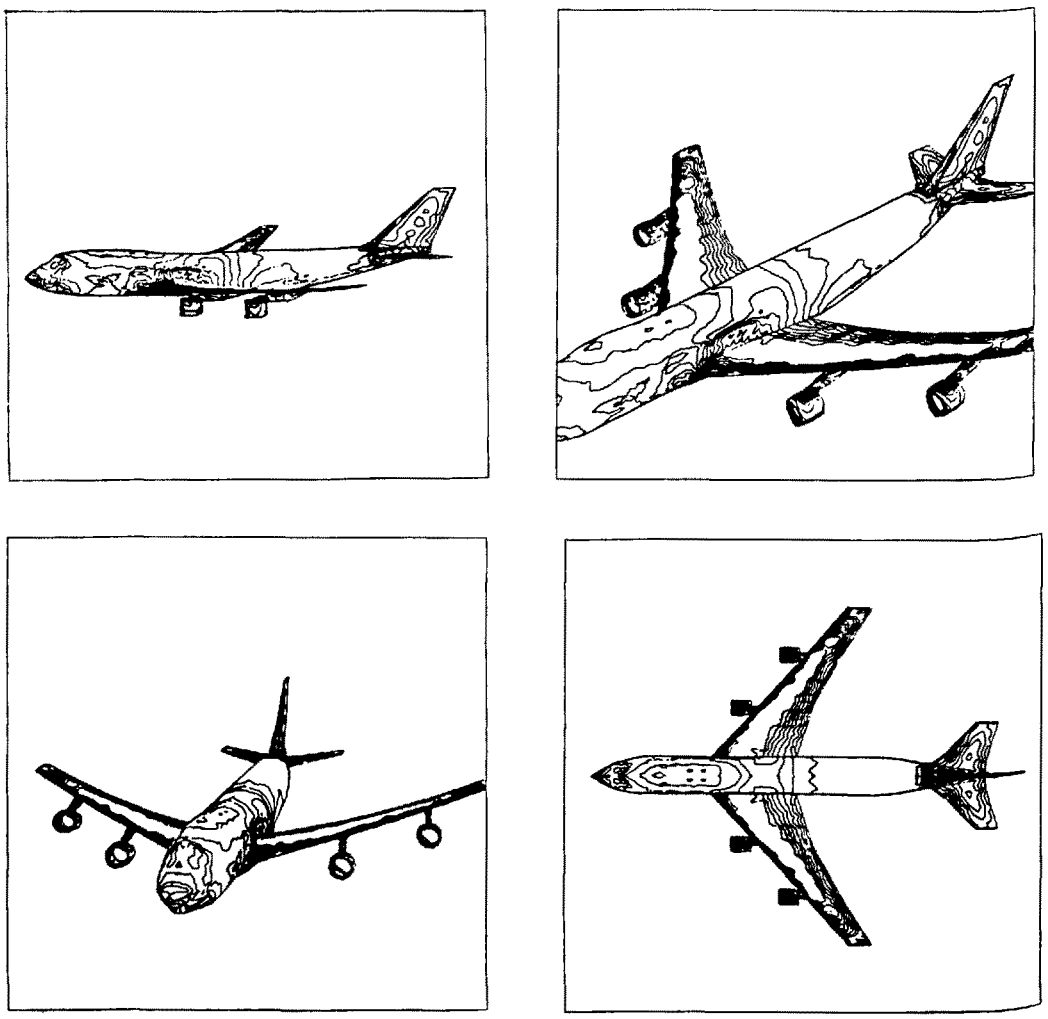

Figure 5. Surface Pressure Contours