CURRENT STATUS AND FUTURE DIRECTIONS

OF COMPUTATIONAL TRANSONICS

Antony Jameson
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544

June 1986

for the ASME Symposium on Future

Directions of Computational Mechanics

Annaheim, December 1986

Abstract

Mathematical models of transonic flow are reviewed and design principles are proposed to guide the development of appropriate numerical methods. Discretization procedures are presented for rectilinear and triangular or tetrahedral meshes. Alternative methods of adding dissipation are discussed, including procedures for constructing total variation diminishing schemes. A variety of explicit and implicit time stepping schemes are reviewed. Trade-offs between efficiency and computational cost are considered, and a general acceleration method using multiple grids is presented. Finally the problem of predicting the flow past a complete aircraft is examined, leading to an assessment of various directions of improvement.

1. Historical Perspective and Guiding Principles

During the last two decades the science of aerodynamics has been transformed by the widespread introduction of computational methods to treat previously intractable problems. The introduction of panel methods in the sixties made it feasible to predict subsonic flows over complex configurations [1-2]. In the seventies and eighties the dominant problem has been the prediction of transonic flow. To a first approximation, cruising efficiency is proportional to the lift to drag ratio multiplied by the speed. This is because the fuel flow of a jet engine does not change much with speed, and also because other costs such as the pay of the crew, maintenance, and depreciation are proportional to time rather than distance. In subsonic flow the attainable lift to drag ratio is roughly constant (in the range of 15 to 20) until compressibility effects become important. Near the speed of sound shock waves appear in the flow, accompanied by drag rise and a sharp decrease in the attainable lift to drag ratio (in the range of 8 for a supersonic transport design). Thus there is a favorable window of operation at high subsonic speeds, just before onset of drag rise. Comparable efficiency is attainable at Mach numbers above three, but this introduces a range of other problems, notably the sonic boom, and the need for materials such as titanium, capable of withstanding high surface temperatures.

The prediction of transonic flow is equally important in the design of military aircraft, since the high lift needed for maneuvers can only be sustained at low supersonic speeds. The richness and complexity of transonic flow have also stimulated the interest of mathematicians [3-4]. A long outstanding question about the existence of shock free transonic flow was finally settled by the

theorem of Morawetz to the effect that shock free transonic flows are isolated solutions [5]. Any small perturbation will cause the appearance of a shock wave. Effective methods were subsequently developed for generating airfoils with shock free design points [6].

My emphasis in this paper is on steady flow. Unsteady flow is inherent in certain other types of flight (that of insects and helicopters, for example), but in the design of aircraft it is important primarily for the analysis of failure modes such as wing flutter. The main objective of computational transonics is the reliable and economical prediction of steady transonic flow past a complete aircraft.

The first choice to be made is that of an appropriate mathematical model. Here there is a trade-off between the accuracy of the prediction and the cost of the calculation. In the seventies major advances were made in the simulation of transonic flow by the small disturbance and potential flow equations [7-11]. Programs such as FL022 [9], and the finite element code developed at Dassault [11], have been widely used to assist the design of commercial transports. With the rapid growth in both speed and memory of the computers now becoming available, the savings in computational costs realizable by the use of these approximations are no longer a dominant factor, and it seems that the future must lie with the full nonlinear equations of fluid flow, which can provide an exact description of shock waves and contact discontinuities. The eighties have in fact seen rapid developments in methods for solving the Euler and Navier-Stokes equations [12-15].

The ultimate requirement will be a solution of the viscous equations. At the Reynolds numbers which prevail in full scale flight, the onset of turbulence

in the boundary layer is inevitable in the absence of some kind of active control. The flow will accordingly be unsteady in the boundary layer. It is clearly not possible to resolve the small scales of turbulence in a calculation for a complete configuration, forcing recourse to statistical averaging and the introduction of turbulence models. No universal turbulence model is available at the present time, and greater accuracy may be attainable by careful matching of an inner viscous solution of the boundary layer with an outer inviscid solution. Improvements in full viscous calculations are likely to be paced by progress in turbulence modelling which may result from better understanding of turbulence [16]. It appears, in any case, that the development of reliable and accurate methods of solving the Euler equations of inviscid flow are an essential first step. This is useful in itself, since the viscous effects are so small outside the boundary layer that an inviscid calculation can provide a valuable insight into the nature of the flow pattern, and help the identification of potential trouble spots, such as excessively strong shock waves which may cause separation. It also lays the foundation for the extension to the Navier Stokes equations, since the numerical approximation of the viscous terms can be accomplished within the same framework, whereas the potential flow models admit no such extension. Accordingly, I shall concentrate in this paper on methods of solving the Euler equations.

Some of the principal difficulties of the problem are:

(1) The equations of gas dynamics are nonlinear.

(2) Solutions in the transonic range will ordinarily be discontinuous: we may expect to find shock waves.

The solutions will also generally contain contact surfaces in the form of vortex sheets.

(3) There are regions of the flow in the neighborhood, for example, of stagnation points, the wing trailing edge or the wing tip, where the derivatives may become very large or even unbounded, leading to large discretization errors.

(4) The equations are to be solved in an unbounded domain.

(5) We are generally interested in calculating flows over bodies of extreme geometric complexity (including cases where the domain is multiply connected).

Assuming that our objective is to calculate steady flow, the introduction of a space discretization procedure reduces the problem to the solution of a large number of coupled nonlinear equations. These equations might be solved by a variety of iterative methods. Two possibilities in particular are the least squares method, which as been successfully employed by Glowinski and his co-workers [11], and the Newton iteration which has recently been used to solve the two dimensional Euler equations by Giles [17]. There are advantages, however, to the strategy of using the time dependent equations as a vehicle for reaching the steady state. Some of these are:

1) Simplicity (with consequent reduction of the risk of programming errors).

2) The possibility of using the same problem to calculate steady and unsteady flows (should the physical problem not have a steady solution the program may then simulate the actual unsteady flow).

3) The time dependent problem provides a suitable framework for the design of non-oscillatory shock capturing schemes which reflect the physics of wave propagation.

4) Algorithms, in particular those using explicit time stepping schemes or iterative procedures at each step of an implicit scheme, can be devised to take maximum advantage of parallel and vector processing.

None of these virtues could be considered decisive if the convergence of a time dependent scheme to a steady state were excessively slow in comparison with competing methods. It turns out, however, that this need not be the case when appropriate acceleration procedures are introduced.

Peter Lax (Baejter Seminar, Princeton, October 1983), has suggested the need for design principles to guide the development of numerical methods for complex scientific problems. Some design principles appropriate to the present problem are listed below.

(1) The conservation laws of gas dynamics should be satisfied in discrete conservation form by the numerical approximation. (We may then rely on the theorem of Lax and Wendroff that the correct shock jump conditions will be satisfied by the solution if it converges in the limit of decreasing mesh width [18]).

(2) Shock waves are to be captured by the introduction of appropriate dissipative terms in the discrete approximation to provide an upwind bias.

(3) The final steady state should be independent of the time marching procedure. (I do not wish to exclude the possibility that the final steady state will depend on the initial state, although there is evidence provided by numerous flying objects suggesting the repeatability of a substantial class of steady solutions). This requirement does, however, exclude the use of a number of popular difference schemes, including schemes with fractional steps, and the Lax Wendroff and MacCormack schemes.

(4) Wherever possible, if a quantity is known to be invariant in the true solution, it should also be invariant in the numerical solution. For example, the total enthalpy should be constant in the steady state solution.

(5) Uniform flow should be an exact solution of the difference equations on an arbitrary mesh.

In order to meet requirement (3) I believe that it is helpful to separate the space discretization procedure entirely from the time marching procedure by applying first a semi-discretization. This has the advantage of allowing the problems of spatial discretization error, artificial dissipation and shock modeling to be studied independently of the problems of time marching stability and convergence acceleration.

The following sections discuss the implementation of these ideas in greater detail. Section 2 reviews alternative space discretization procedures on both rectilinear and triangular meshes. These are presented in the context of finite

volume methods, but some of them can equally well be interpreted as finite element methods. Section 3 discusses dissipation, its connection with upwinding, and the construction of total variation diminishing schemes. Time discretization is discussed in Section 4, and acceleration by the use of multiple grids in Section 5. Finally Sections 6 and 7 review the problem of computing the flow past a complete aircraft, and point the way toward some future developments.

2. Space Discretization of the Euler Equations

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

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

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

$$\frac{\partial}{\partial t}\iiint_{\Omega} w d\Omega + \iint_{\partial\Omega} \underline{F} \cdot d\underline{S} = 0 \tag{2.1}$$

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

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

For conservation of momentum in the x direction

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

with similar definitions for the y and z directions, and for energy conservation

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

If we divide the domain into a large number of small subdomains, we can use equation (2.1) to estimate the average rate of change of w in each subdomain. This is an effective method to obtain discrete approximations to equation (2.1) which preserve its conservation form. In general the subdomains could be arbitrary, but it is convenient to use either distorted cubic or tetrahedral cells. Alternative discretizations may be obtained by storing sample values of

the flow variables at either the cell centers or the cell corners. These variations are illustrated in Figure 1 for a two-dimensional case.

Figures 1(a) and 1(b) show cell centered schemes on rectilinear and triangular meshes [12,19]. In either case equation (1) is written for the cell labelled 0 as

$$\frac{d}{dt}(Vw) + Q = 0 \tag{2.2}$$

where V is the cell volume and Q is the net flux out of the cell. This can be approximated as

$$Q = \sum_k \underline{F}_{0k} \cdot \underline{S}_{0k} \tag{2.3}$$

where the sum is over the faces of cell 0, $\underline{S}_{0k}$ is the directed area of the face separating cell 0 from cell k, and the flux $\underline{F}_{0k}$ is evaluated by taking the average of its value in cell 0 and cell k.

$$\underline{F}_{0k} = \frac{1}{2}(\underline{F}_0 + \underline{F}_k) \tag{2.4}$$

An alternative averaging procedure is to multiply the average value of the convected quantity, ρ_{0k} in the case of the continuity equation, for example, by the transport vector

$$Q_{0k} = \frac{1}{2}(\underline{q}_0 + \underline{q}_k) \cdot \underline{S}_{0k} \tag{2.4*}$$

obtained by taking the inner product of the mean of the velocity vector $\underline{q}$ with the directed face area.

Figures 1(c) and 1(d) show corresponding schemes on rectilinear and triangular meshes in which the flow variables are stored at the vertices. We can now form a control volume for each vertex by taking the union of the cells meeting at

that vertex. Equation (2.1) then takes the form

$$\frac{d}{dt} \left(\sum_k V_k \right) w + \sum_k Q_k = 0 \tag{2.5}$$

where V_k and Q_k are the cell volume and flux balance for the kth cell in the control volume. The flux balance for a given cell is now approximated as

$$Q = \sum_\ell \underline{F}_\ell \cdot \underline{S}_\ell \tag{2.6}$$

where $\underline{S}_\ell$ is the directed area of the ℓth face, and $\underline{F}_\ell$ is an estimate of the mean flux vector across that face. Fluxes across internal faces cancel when the sum $\sum_k Q_k$ is taken in equation (2.5), so that only the external faces of the control volume contribute to its flux balance.

In the two dimensional case the mean flux across an edge can be conveniently approximated as the average of the values at its two end points,

$$\underline{\bar{F}}_{12} = \frac{1}{2} (\underline{F}_1 + \underline{F}_2)$$

in Figure 1(c) or 1(d), for example. The sum $\sum Q_k$ in equation (2.5), which then amounts to a trapezoidal integration rule around the boundary of the control area, should remain fairly accurate even when the mesh is irregular. This is an advantage of the vertex formulation over the cell centered formulation, in which the midpoint of the line joining the sample values does not necessarily coincide with the midpoint of the corresponding edge, with a consequent reduction of accuracy on a distorted or kinked mesh (see Figure 2).

Storage of the solution at the vertices has a similar advantage when a tetrahedral mesh is used in a three dimensional calculation [20,21]. The use of a simple average of the three corner values of each triangular face

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

is a natural choice, which is consistent with the assumption that $\underline{F}$ varies linearly over the face. The following argument shows that the resulting scheme is essentially equivalent to the use of a Galerkin method with piecewise linear basis functions. Consider the differential form of equation (2.1)

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

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

$$\frac{\partial}{\partial t} \iiint_{\Omega} \phi w d\Omega = \iiint_{\Omega} \underline{F} \cdot \nabla\phi d\Omega - \iint_{\partial\Omega} \phi \underline{F} \cdot d\underline{S} \qquad (2.7)$$

Suppose now that we take ϕ to be the piecewise linear function with the value unity at one node (denoted by 0 in Figure 3), 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

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

where V is the cell volume, S_{x_k} and $\overline{\phi}_k$ 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 $\overline{\phi} = 1/3$ 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

$$\nabla\phi = -\underline{S}/3V \qquad (2.8)$$

where $\underline{S}$ is the directed area of the face opposite vertex 0. Now treat $\underline{F}$ as piecewise linear and use equation (2.8) to evaluate the volume integral on the

right side of equation (2.7). Then each tetrahedron meeting at node 0 introduces a contribution $(\overline{\underline{F}} \cdot \underline{S})/3$ where $\overline{\underline{F}}$ is the average value of $\underline{F}$ in the cell. For the cell illustrated in Figure 1, for example,

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

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

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

Since the control volume is closed, however,

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

Therefore the contribution of F_0 to $\overline{F}_k$ 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 (2.7) can be replaced by

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

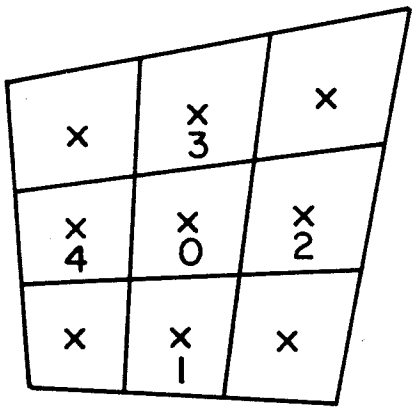
On the left hand side of equation (2.7) we take w to be constant inside the control volume. Since ϕ is piecewise linear, the volume average value is $\overline{\phi} = 1/4$. The factor 1/4 cancels on each side and the approximation to equation (2.7) can therefore be written as

$$\frac{d}{dt}\left(\sum_k V_k\right) w + \sum_k \tilde{\underline{F}}_k \cdot \underline{S}_k = 0 \qquad (2.9)$$

which is equivalent to equation (2.5)

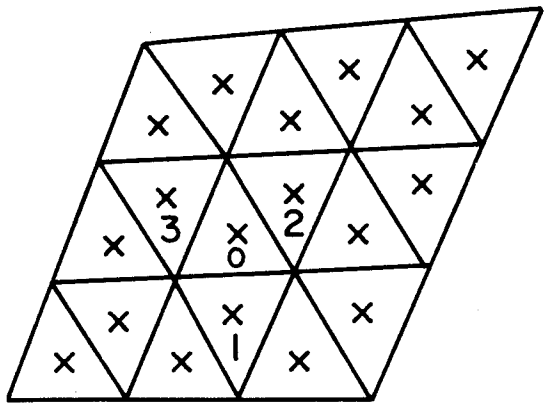
Referring to Figure 4, which illustrates a two dimensional mesh, it may be

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 5. Vertices 4 and 5 are the centers of the two control volumes influenced by the face. It is now possible to generate the approximation (2.9) 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.



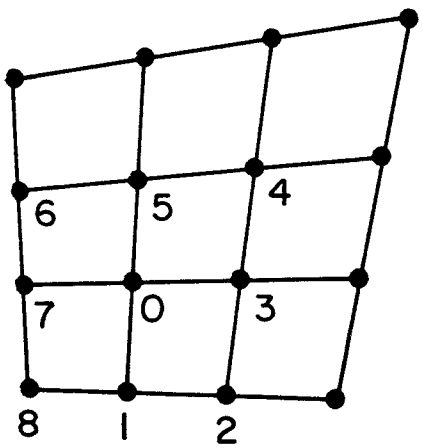
(a)

CELL CENTERED RECTILINEAR



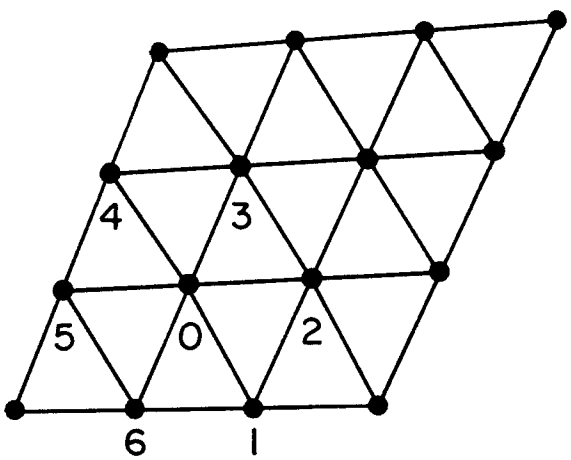
(b)

CELL CENTERED TRIANGULAR



(c)

VERTEX RECTILINEAR



(d)

VERTEX TRIANGULAR

Figure 1

Alternative discretization schemes

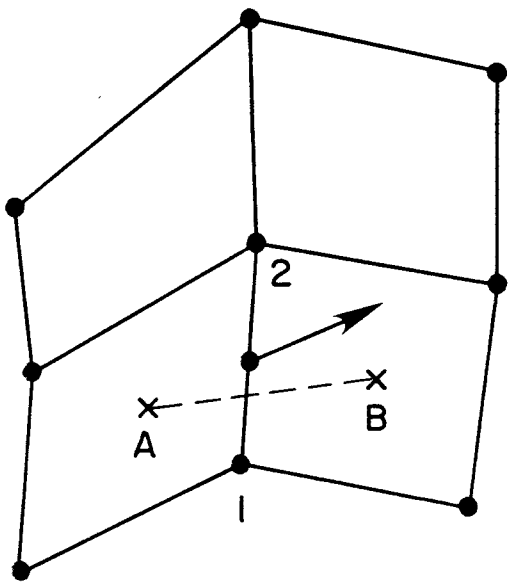


Figure 2

Comparison of discretization schemes on a kinked mesh.
Evaluation of $\underline{F}$ at P by averaging $\underline{F}_1$ and $\underline{F}_2$
is more accurate that averaging $\underline{F}_A$ and $\underline{F}_B$.

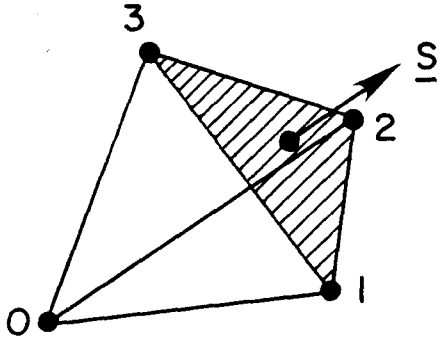


Figure 3

One tetrahedron of the control volume centered at node 0.

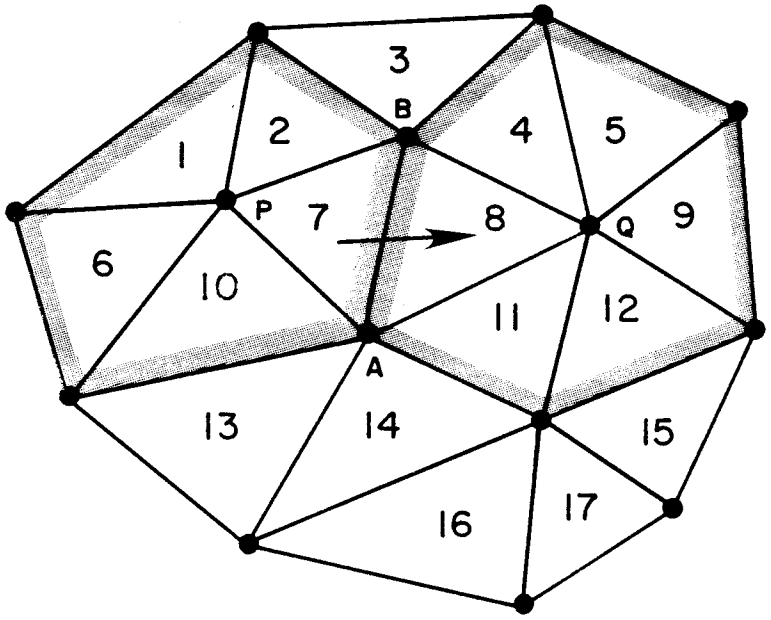


Figure 4

A triangular mesh in 2 directions: 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, 9 and 5. The flux across the edge AB is from the control volume at P to the control volume at Q.

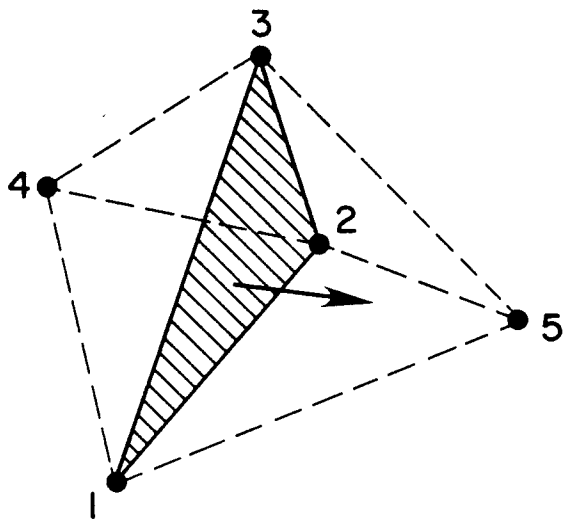


Figure 5

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.

3. Dissipation, Upwinding and Total Variation Diminishing Schemes

Equations (2.2) or (2.5) represent nondissipative approximations to the Euler equations. Dissipative terms may be needed for two reasons. First there is the possibility of undamped oscillatory modes. For example, when either a cell centered or a vertex formulation is used to represent a conservation law on a rectilinear mesh, a mode with values ± 1 alternately at odd and even points leads to a numerically evaluated flux balance of zero in every interior control volume. Although the boundary conditions may suppress such a mode in the steady state solution, the absence of damping at interior points may have an adverse effect on the rate of convergence to the steady state.

The second reason for introducing dissipative terms is to allow the clean capture of shock waves and contact discontinuities without undesirable oscillations. Following the pioneering work of Godunov [22], a variety of dissipative and upwind schemes designed to have good shock capturing properties have been developed during the past decade [23-33]. The one-dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \tag{3.1}$$

provides a useful model for the analysis of these schemes. The total variation

$$TV = \int_{-\infty}^{\infty} \left| \frac{\partial u}{\partial x} \right| dx$$

of a solution of (3.1) does not increase, provided that any discontinuity appearing in the solution satisfies an entropy condition [34]. The concept of total variation diminishing (TVD) difference schemes, introduced by Harten [28], provides a unifying framework for the study of shock capturing methods. These are schemes with the property that the total variation of the discrete solution

$$TV = \sum_{-\infty}^{\infty} \left| v_j - v_{j-1} \right|$$

cannot increase. The general conditions for a multipoint one-dimensional scheme to be TVD have been stated and proved by Jameson and Lax [35]. For a semi-discrete scheme expressed in the form

$$\frac{d}{dt} v_j = \sum_{j=-Q}^{Q-1} c_q(j)(v_{j-q} - v_{j-q-1}) \tag{3.2}$$

these conditions are

$$c_{-1}(j-1) \geq c_{-2}(j-2) \cdots \geq c_{-j}(j-Q) \geq 0 \tag{3.3a}$$

and

$$-c_0(j) \geq -c_1(j+1) \cdots \geq -c_{Q-1}(j+Q-1) \geq 0 \tag{3.3b}$$

Specialized to a three point scheme these conditions imply that the scheme

$$\frac{d}{dt} v_j = c_{j+1/2}(v_{j+1} - v_j) - c_{j-1/2}(v_j - v_{j-1})$$

is TVD if $c_{j+1/2} \geq 0$, $c_{j-1/2} \geq 0$.

A conservative semi-discrete approximation to equation (3.1) can be derived by subdividing the line into cells. Then the evolution of the value v_j in the jth cell is given by

$$\Delta x \frac{d}{dt} v_j + h_{j+1/2} - h_{j-1/2} = 0 \tag{3.4}$$

where $h_{j+1/2}$ is the estimate of the flux between cells j and j + 1. Conditions (3.3) are satisfied by the upwind scheme

$$h_{j+1/2} = \begin{cases} f(v_j) & \text{if } a_{j+1/2} \geq 0 \\ f(v_{j+1}) & \text{if } a_{j+1/2} < 0 \end{cases} \tag{3.5}$$

where $a_{j+1/2}$ is a numerical estimate of the wave speed $a = \partial f/\partial u$,

$$a_{j+1/2} = \begin{cases} \dfrac{f_{j+1} - f_j}{v_{j+1} - v_j} & \text{if } v_{j+1} \neq v_j \\ \left.\dfrac{\partial f}{\partial v}\right|_{v = v_j} & \text{if } v_{j+1} = v_j \end{cases} \tag{3.6}$$

More generally, if one sets

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) + \alpha_{j+1/2}(v_{j+1} - v_j) \tag{3.7}$$

where $\alpha_{j+1/2}$ is a dissipative coefficient, the scheme is TVD if

$$\alpha_{j+1/2} \geq \frac{1}{2}\left| a_{j+1/2} \right| \tag{3.8}$$

since one can write

$$h_{j+1/2} = f_j + \frac{1}{2}(f_{j+1} - f_j) - \alpha_{j+1/2}(v_{j+1} - v_j)$$

$$= f_j + (\frac{1}{2} a_{j+1/2} - \alpha_{j+1/2})(v_{j+1} - v_j)$$

and

$$h_{j-1/2} = f_j - \frac{1}{2}(f_j - f_{j-1}) - \alpha_{j-1/2}(v_j - v_{j-1})$$

$$= f_j - (\frac{1}{2} a_{j-1/2} + \alpha_{j-1/2})(v_j - v_{j-1})$$

Thus the use of a dissipative coefficient with a magnitude of at least half the wave speed produces a TVD scheme, while the minimum sufficient value produces the upwind scheme.

TVD schemes preserve the monotonicity of an intially monotone profile, because the total variation would increase if the profile ceased to be monotone. Consequently, they prevent the formation of spurious oscillations. In this simple form, however, they are at best first order accurate. Harten devised a second order accurate TVD scheme by introducing antidiffusive terms with flux limiters [28]. The use of antidiffusive terms and flux limiters to improve shock resolution can be traced to the work of Boris and Book [23]. The concept of the flux limiting was independently advanced by Van Leer [24].

A particularly simple method of constructing a second order accurate TVD scheme is to introduce flux limiters directly into a higher-order dissipative term [33]. Define the numerical flux in equation (3.4) as

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) + d_{j+1/2} \tag{3.9}$$

where $d_{j+1/2}$ is a dissipative flux. Suppose that this is constructed as

$$d_{j+1/2} = e_{j+3/2} - 2e_{j+1/2} + e_{j-1/2} \tag{3.10}$$

where

$$e_{j+1/2} = \alpha_{j+1/2}(v_{j+1} - v_j) \tag{3.11}$$

and $\alpha_{j+1/2}$ is a positive coefficient. According to equations (3.10) and (3.11) the dissipative flux is a quantity of third order. Now equation (3.4) becomes

$$\Delta x \frac{dv_j}{dt} = -\frac{1}{2} a_{j+1/2} \Delta_{j+1/2} - \frac{1}{2} a_{j-1/2} \Delta_{j-1/2}$$

$$- \alpha_{j+3/2} \Delta_{j+3/2} + 3\alpha_{j+1/2} \Delta_{j+1/2} - 3\alpha_{j-1/2} \Delta_{j-1/2} + \alpha_{j-3/2} \Delta_{j-3/2}$$

where

$$\Delta_{j+1/2} = v_{j+1} - v_j$$

This does not satisfy conditions (3.3) because the coefficients of $\Delta_{j+3/2}$ and $\Delta_{j-3/2}$ have the wrong sign. In order to correct this we can modify the dissipative terms by the introduction of flux limiters. Denote the ratio of successive increments by

$$r_j = \frac{e_{j-1/2}}{e_{j+1/2}} \tag{3.12}$$

and define the function

$$\phi(r) = \begin{cases} 0, & r < 0 \\ r, & 0 \leq r \leq 1 \\ 1, & r > 1 \end{cases} \tag{3.13}$$

Also let

$$\psi(r) = \phi(\frac{1}{r}) \tag{3.14}$$

Since ϕ satisfies the symmetry condition

$$r\ \phi(\frac{1}{r}) = \phi(r)$$

it follows that

$$r\ \psi(r) = \phi(r) \leq 1$$

Denoting $\phi(r_j)$ by ϕ_j and $\psi(r_j)$ by ψ_j the dissipative flux is now redefined as

$$d_{j+1/2} = \phi_{j+1}\ e_{j+3/2} - 2e_{j+1/2} + \psi_j\ e_{j-1/2} \tag{3.15}$$

According to equation (3.12)

$$e_{j+3/2} = \frac{e_{j+1/2}}{r_{j+1}}\ ,\ e_{j-3/2} = r_{j-1}\ e_{j-1/2}$$

Therefore equation (3.4) now yields

$$\Delta x\ \frac{dv_i}{dt} = -\frac{1}{2}\ a_{j+1/2}\ \Delta_{j+1/2} - \frac{1}{2}\ a_{j-1/2}\ \Delta_{j-1/2}$$

$$+ (2 - \frac{\phi_{j+1}}{r_{j+1}} + \phi_j)\ \alpha_{j+1/2}\ \Delta_{j+1/2}$$

$$- (2 - r_{j-1}\ \psi_{j-1} + \psi_j)\ \alpha_{j-1/2}\ \Delta_{j-1/2}$$

Since

$$0 \leq \phi(r) \leq 1 \quad , \quad 0 \leq \frac{\phi(r)}{r} \leq 1$$

and

$$0 \leq \psi(r) \leq 1 \quad , \quad 0 \leq r\,\psi(r) \leq 1$$

it follows that conditions (3.3) are satisfied if condition (3.8) holds. In a region where the solution is smooth

$$r_j = 1 + 0(\Delta x)$$

with the result that

$$\phi_j = 1 + 0(\Delta x) \ , \ \psi_j = 1 + 0(\Delta x)$$

and the modification of equation (3.10) is of second order.

A convenient way of applying these ideas to a system of equations was proposed by Roe [26]. Let $A_{j+1/2}$ be a matrix with the property that the flux difference satisfies the relation

$$f(w_{j+1}) - f(w_j) = A_{j+1/2}(w_{j+1} - w_j) \tag{3.16}$$

Roe gives a method for the construction of such a matrix, which is a numerical approximation to the Jacobian matrix $\partial f/\partial w$. Its eigenvalues λ_ℓ are thus numerical estimates of the wave speeds associated with the system. Now decompose the difference $w_{j+1} - w_j$ as a sum of the eigenvectors r_ℓ of $A_{j+1/2}$,

$$w_{j+1} - w_j = \sum \alpha_\ell \, r_\ell \tag{3.17}$$

Then

$$f_{j+1} - f_j = \sum \lambda_\ell \, \alpha_\ell \, r_\ell \tag{3.18}$$

and the desired dissipative term can be constructed as

$$\sum \mu_\ell \alpha_\ell r_\ell \tag{3.19}$$

where

$$\mu_\ell \geq \frac{1}{2} \left| \lambda_\ell \right| \tag{3.20}$$

This method amounts to constructing separate dissipative terms for the characteristic variables defined by the eigenvectors of $A_{J+1/2}$. It is closely related to the concept of flux splitting first introduced by Steger and Warming [25], in which the flux vector itself is split into components corresponding to the wave speeds, and backward differencing is used for the part propagating forwards, while forward differencing is used for the part propagating backwards. Alternative methods of flux splitting which lead to excellent shock capturing schemes have been proposed by Osher [27] and Van Leer [31].

These concepts can be applied to two and three dimensional problems by separately applying the one-dimensional construction in each coordinate direction. There is no theoretical basis for this, but it generally leads to good results in practice. The cell centered finite volume formulation is readily adapted to this kind of construction. A first order upwind scheme can be constructed by splitting the flux across each face into components corresponding to forward and backward propagation, and then evaluating each component by taking values from the cell on the upwind side of the face.

An alternative approach is as follows. Consider a two-dimensional scalar conservation law of the form

$$\frac{\partial v}{\partial f} + \frac{\partial}{\partial x} f(v) + \frac{\partial}{\partial y} g(v) = 0 \tag{3.21}$$

The mesh may be either rectilinear or triangular, as sketched in Figure 1. Assume that the evolution equation at the mesh point 0 depends on contributions from the nearest neighbors, numbered as in the figure. Suppose that this is expressed in the form

$$\frac{dv_0}{dt} = \sum_k c_k(v_k - v_0) \tag{3.22}$$

where the sum is over the neighbors. Then we require all the coefficients to be nonnegative

$$c_k \geq 0, \quad k = 1,2... \tag{3.23}$$

This condition on the signs of the coefficients, which is a direct generalization of the conditions for a one dimensional three point scheme to be TVD, assures that a maximum cannot increase. Finite volume approximations to equation (3.21) can be reduced to the form (3.22) by making use of the fact that the sums $\sum_k \Delta x$ and $\sum_k \Delta y$ taken around the perimeter of the control area are zero, so that a multiple of $f(v_0)$ or $g(v_0)$ can be subtracted from the flux. Consider, for example, a formulation in which v is stored at the vertices of a triangular mesh, as in Figure 1(d). Then equation (3.21) is replaced by

$$S\frac{dv_0}{dt} + \frac{1}{2}\sum_k \{(f_k + f_{k-1})(y_k - y_{k-1}) - (g_k + g_{k-1})(x_k - x_{k-1})\} = 0 \tag{3.24}$$

where k ranges from 1 to 6, and S is the area of the polygon. This can be rearranged as

$$S\frac{dv_0}{dt} + \sum_k \{f(v_k)\Delta y_k - g(v_k)\Delta x_k\} = 0$$

where

$$\Delta x_k = \frac{1}{2}(x_{k+1} - x_{k-1}), \quad \Delta y_k = \frac{1}{2}(y_{k+1} - y_{k-1}),$$

and this is equivalent to

$$S\frac{dv_0}{dt} + \sum_k \{(f(v_k) - f(v_0))\Delta y_k + (g(v_k) - g(v_0))\Delta x_k\} = 0 \quad (3.25)$$

Define the coefficient a_{k0} as

$$a_{k0} = \begin{cases} \dfrac{(f_k - f_0)\Delta y_k - (g_k - g_0)\Delta x_k}{v_k - v_0}, & v_k \neq v_0 \\ \left(\dfrac{\partial f}{\partial v}\Delta y_k - \dfrac{\partial g}{\partial v}\Delta x_k\right)\Big|_{v = v_0}, & v_k = v_0 \end{cases} \quad (3.26)$$

Then equation (3.25) reduces to

$$S\frac{dv_0}{dt} + \sum_k a_{k0}(v_k - v_0) = 0.$$

To produce a scheme satisfying the sign condition (3.23), add a dissipative term on the right hand side of the form

$$\sum_k \alpha_{k0}\,(v_k - v_0)$$

where the coefficients α_{k0} satisfy the condition

$$\alpha_{k0} \geq |\, a_{k0} \,| \quad (3.27)$$

The definition (3.26) and condition (3.27) correspond to the definition (3.6) and condition (3.8) in the one dimensional case. The extension to a system can be carried out with the aid of Roe's construction. Now a_{k0} is replaced by the corresponding matrix A_{k0} such that

$$A_{k0}(w_k - w_0) = (f_k - f_0)\,\Delta y_k - (g_k - g_0)\,\Delta x_k$$

Then $w_k - w_0$ is expanded as a sum of the eigenvectors of A_{k0}, and a contribution to the dissipative term is formed by multiplying each eigenvector by a positive coefficient with a magnitude not less than that of the corresponding eigenvalue.

The use of flux splitting allows precise matching of the dissipative terms to introduce the minimum amount of dissipation needed to prevent oscillations.

This in turn reduces the thickness of the numerical shock layer to the minimum attainable, one or two cells for a normal shock. In practice, however, it turns out that shock waves can be quite cleanly captured without flux splitting by using adaptive coefficients. The dissipation then has a low background level which is increased in the neighborhood of shock waves to a peak value proportional to the maximum local wave speed. The second difference of the pressure has been found to be an effective measure for this purpose. The dissipative terms are constructed in a similar manner for each dependent variable by introducing dissipative fluxes which preserve the conservation form.

For a three dimensional rectilinear mesh the added terms have the form

$$d_{i+1/2,j,k} - d_{i-1/2,j,k} + d_{i,j+1/2,k} - d_{i,j-1/2,k} + d_{i,j,k+1/2} - d_{i,j,k-1/2} \tag{3.28}$$

These fluxes are constructed by blending first and third differences of the dependent variables. For example, the dissipative flux in the i direction for the mass equation is

$$d_{i+1/2,j,k} = R(\varepsilon^{(2)} - \varepsilon^{(4)} \delta_x^2)(\rho_{i+1,j,k} - \rho_{i,j,k}) \tag{3.29}$$

where δ_x^2 is the second difference operator, $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are the adaptive coefficients, and R is a scaling factor proportional to an estimate of the maximum local wave speed. For an explicit scheme the local time step limit Δt^* is a measure of the time it takes for the fastest wave to cross a mesh interval, and R can accordingly by made proportional to $1/\Delta t^*$. The coefficient $\varepsilon^{(4)}$ provides the background dissipation in smooth parts of the flow, and can be used to improve the capability of the scheme to damp high frequency modes. Shock cap-

turing is controlled by the coefficient $\varepsilon^{(2)}$, which is made proportional to the normalized second difference of the pressure

$$\nu_{i,j,k} = \left| \frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}} \right|$$

in the adjacent cells.

The realization of dissipative terms of this type on a triangular mesh is illustrated in Figure 6. 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

$$D_0 = \sum_k \varepsilon^{(1)}_{k0}(w_k - w_0) \tag{3.30}$$

where the sum is over the nearest neighbors. 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. A more accurate scheme is obtained by recycling the edge differencing procedure. After first setting

$$E_0 = \sum_k (w_k - w_0) \tag{3.31}$$

at every mesh point, one then sets

$$D_0 = - \sum_k \varepsilon^{(2)}_{0k} (E_k - E_0) \tag{3.32}$$

An effective scheme is produced by blending formulas (3.30) and (3.32), 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). The required sums can be efficiently assembled by loops over the edges.

Schemes constructed along these lines combine the advantages of simplicity and economy of computation, at the expense of an increase in thickness of the numerical shock layer to three or four cells. They have also proved robust in calculations over a wide range of Mach numbers (extending up to 20 in recent studies [36]).

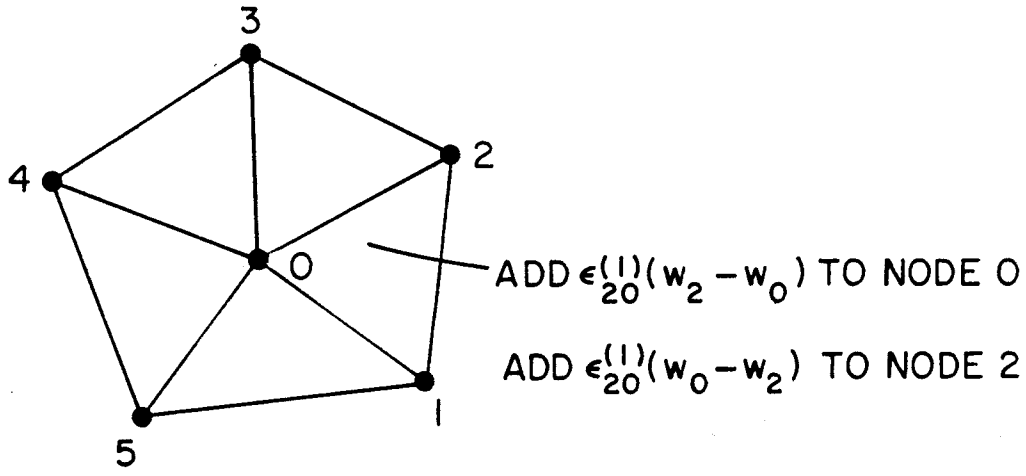


Figure 6

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

4. Time Stepping Schemes

The discretization procedures of Section 2 lead to a set of coupled ordinary differential equations, which can be written in the form

$$\frac{dw}{dt} + R(w) = 0 \tag{4.1}$$

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 equations (2.2) or (2.5), together with the added dissipative terms. These are to be integrated to a steady state. Since the objective is simply to reach the steady state and details of the transient solution are immaterial, the time stepping scheme may be designed solely to maximize the rate of convergence without having to meet any constraints imposed by the need to achieve a specified level of accuracy, provided that it does not interfere with the definition of the residual R(w). Figure 7 indicates some of the principal time stepping schemes which might be considered. The first major choice is whether to use an explicit or an implicit scheme.

A very widely used explicit scheme is that proposed by MacCormack [37], which uses the predictor and corrector steps

$$w^{*} = w^{n} - \Delta t(D_x^{+} f^{n} + D_y^{+} g^{n}) \tag{4.2a}$$

and

$$w^{n+1} = w^{n} - \frac{\Delta t}{2}(D_x^{+} f^{n} + D_y^{+} g^{n}) - \frac{\Delta t}{2}(D_x^{-} f^{*} + D_y^{-} g^{*}) \tag{4.2b}$$

where D_x^{+} , D_x^{-} , D_y^{+} and D_y^{-} are forward and backward difference operators approximating $\partial/\partial x$ and $\partial/\partial y$. This minimizes the number of computer operations needed to realize a second order accurate scheme. The use of different approxi-

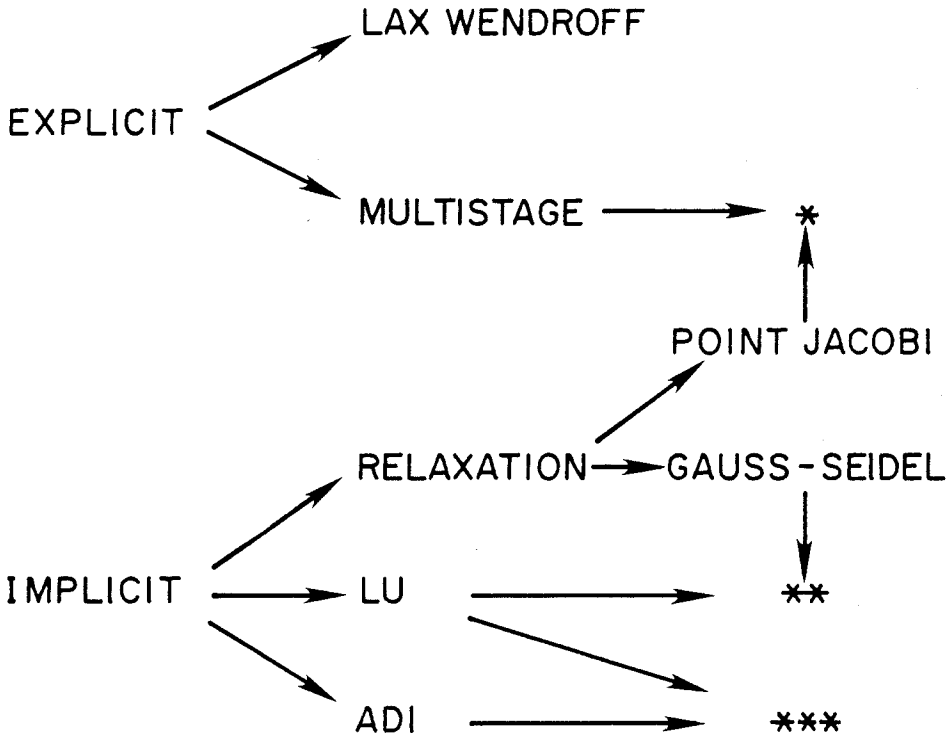


Figure 7

Time Stepping Schemes

* facilitates vector and parallel processing

mations for $\partial f/\partial x + \partial g/\partial y$ in the two stages leads, however, to a dependence of the steady state solution on Δt.

If one regards equation (4.1) as a set of ordinary differential equations in which R(w) has a fixed form, then the steady state solution is unambiguously R(w) = 0. Explicit schemes which might be considered include linear multistep methods such as the leap frog and Adams-Bashforth schemes, and one step multistage methods such as the classical Runge-Kutta schemes. The one step multistage schemes have the advantages that they require no special start up procedure, and that they can readily be tailored to give a desired stability region. They have proved extremely effective in practice as a method of solving the Euler equations.

Let w^n be the result after n steps. The general form of an m stage scheme is

$$\begin{aligned} w^{(0)} &= w^n \\ w^{(1)} &= w^{(0)} - \alpha_1 \Delta t\, R^{(0)} \\ &\cdots \\ w^{(m-1)} &= w^{(0)} - \alpha_{m-1} \Delta t\, R^{(m-2)} \\ w^{(m)} &= w^{(0)} - \Delta t\, R^{(m-1)} \\ w^{n+1} &= w^{(m)} \end{aligned} \tag{4.3}$$

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

$$R^{(q)} = \sum_{r=0}^{q} \beta_{qr} R(w^{(r)}) \tag{4.4}$$

where

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

In the simplest case

$$R^{(q)} = R(w^{(q)})$$

It is then known how to choose the coefficients α_q to maximize the stability interval along the imaginary axis, and consequently the time step [38]. Since only the steady state solution is needed, it pays to separate the residual R(w) into its convective and dissipative parts Q(w) and D(w). Then the residual in the (q+1)st stage is evaluated as

$$R^{(q)} = \sum_{r=0}^{q} \{\beta_{qr} Q(w^{(r)}) - \gamma_{qr} D(w^{(r)})\} \qquad (4.4^*)$$

where

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

Blended multi-stage schemes of this type, which have been analyzed in reference 39, can be tailored to give large stability intervals along both the imaginary and negative real axes.

The properties of multi-stage schemes can be further enhanced by residual averaging [13]. Here the residual at a mesh point is replaced by a weighted average of neighboring residuals. The average is calculated implicitly. In a one dimensional case R(w) is replaced by $\bar{R}(w)$, where at the jth mesh point

$$-\varepsilon \bar{R}_{j-1} + (1+2\varepsilon)\bar{R}_j - \varepsilon \bar{R}_{j+1} = R_j$$

It can easily be shown that the scheme can be stabilized for an arbitrarily large time step by choosing a sufficiently large value for ε. In a non-dissipative one dimensional case one needs

$$\varepsilon \geq \frac{1}{4} \left(\left(\frac{\Delta t}{\Delta t^*}\right)^2 - 1\right)$$

where Δt^* is the maximum stable time step of the basic scheme, and Δt is the actual time step. The method can be extended to three dimensions by using smoothing in product form

$$(1 - \varepsilon_x \delta_x^2)(1 - \varepsilon_y \delta_y^2)(1 - \varepsilon_z \delta_z^2)\, \bar{R} = R \tag{4.5}$$

where δ_x^2, δ_y^2 and δ_z^2 are second difference operators in the coordinate directions, and ε_x, ε_y and ε_z are the corresponding smoothing coefficients. Residual averaging can also be used on triangular meshes [19,20]. The implicit equations are then solved by a Jacobi iteration.

One can anticipate that implicit schemes will yield convergence in a smaller number of time steps, since the time step is no longer constrained by a stability limit. This will only pay, however, if the decrease in the number of time steps outweighs the increase in the computational effort per time step consequent upon the need to solve coupled equations. The prototype implicit scheme can be formulated by estimating $\partial w/\partial t$ at $t + \mu\Delta t$ as a linear combination of $R(w^n)$ and $R(w^{n+1})$. The resulting equation

$$w^{n+1} = w^n - \Delta t\{(1-\mu)\, R(w^n) + \mu R(w^{n+1})\} \tag{4.6}$$

can be linearized as

$$(I + \mu\Delta t \frac{\partial R}{\partial w})\, \delta w + \Delta t\, R(w^n) = 0 \tag{4.7}$$

Equation (4.7) reduces to the Newton iteration if one sets $\mu = 1$ and lets $\Delta t \to \infty$. In a three dimensional case with an NxNxN mesh its bandwidth is of order N^2. Direct inversion requires a number of operations proportional to the number of unknowns multiplied by the square of the bandwidth, that is $O(N^7)$. This is prohibitive, and forces recourse to either an approximate factorization method or an iterative solution method.

The main possibilities for approximate factorization are the alternating direction and LU decomposition methods. The alternating direction method, which may be traced back to the work of Gourlay and Mitchell [40], was given an elegant formulation for nonlinear problems by Beam and Warming [41]. In a two dimensional case equation (3.7) is replaced by

$$(I + \mu\Delta t D_x A)(I + \mu\Delta t D_y B)\delta w + \Delta t\, R(w) = 0 \qquad (4.8)$$

where D_x and D_y are difference operators approximating $\partial/\partial x$ and $\partial/\partial y$, and A and B are the Jacobian matrices. This may be solved in two steps:

$$(1) \quad (I + \mu\Delta t D_x A)\delta w^* = -\Delta t\, R(w)$$

$$(2) \quad (I + \mu\Delta t D_y B)\delta w = \delta w^*$$

Each step requires block tridiagonal inversions, and may be performed in $O(N^2)$ operations on an N x N mesh. The algorithm is amenable to vectorization by simultaneous solution of the tridiagonal equations along parallel coordinate lines. The method has been refined to a high level of efficiency by Pulliam and Steger [14], and Yee has extended it to incorporate a TVD scheme [32]. Its main disadvantage is that its extension to three dimensions is inherently unstable according a Von Neumann analysis.

The idea of the LU decomposition method [42] is to replace the operator in equation (4.3) by the product of lower and upper block triangular factors L and U,

$$LU\ \delta w + \Delta t\ R(w) = 0 \tag{4.9}$$

Two factors are used independent of the number of dimensions, and the inversion of each can be accomplished by inversion of its diagonal blocks. The method can be conveniently illustrated by considering a one dimensional example. Let the Jacobian matrix $A = \partial f/\partial w$ be split as

$$A = A^+ + A^-$$

where the eigenvalues of A^+ and A^- are positive and negative, respectively. Then we can take

$$L \equiv I + \mu\Delta t\ D_x^-\ A^+\ ,\ U \equiv I + \mu\Delta t\ D_x^+\ A^- \tag{4.10}$$

where D_x^+ and D_x^- denote forward and backward difference operators approximating $\partial/\partial x$. The reason for splitting A is to ensure the diagonal dominance of L and U, independent of Δt. Otherwise stable inversion of both factors will only be possible for a limited range of Δt. A crude choice is

$$A^{\pm} = \frac{1}{2}\ (A \pm \rho I)$$

where ρ is at least equal to the spectral radius of A. If flux splitting is used in the calculation of the residual, it is natural to use the corresponding splitting for L and U. An interesting variation is to combine an alternating direction scheme with LU decomposition in the different coordinate directions [43,44].

If one chooses to adopt the iterative solution technique, the principal alternatives are variants of the Gauss-Seidel and Jacobi methods. These may be

applied to either the nonlinear equation (4.6) or the linearized equation (4.7). A Jacobi method of solving (4.6) can be formulated by regarding it as an equation

$$w - w^{(0)} + \mu\Delta t\ R(w) + (1-\mu)\ \Delta t\ R(w^{(0)}) = 0$$

to be solved for w. Here $w^{(0)}$ is a fixed value obtained as the result of the previous time step. Such a procedure is a variant of the multi-stage time stepping scheme described by equations (4.3) and (4.4). It has the advantage that it permits simultaneous or overlapped calculation of the corrections at every mesh point, and is readily amenable to parallel and vector processing.

A symmetric Gauss-Seidel scheme has been successfully employed in several recent works[15,45]. Consider the case of a flux split scheme in one dimension, for which

$$R(w) = D_x^+ f^-(w) + D_x^- f^+(w)$$

where the flux is split so that the Jacobian matrices

$$A^+ = \frac{\partial f^+}{\partial w} \quad \text{and} \quad A^- = \frac{\partial f^-}{\partial w}$$

have positive and negative eigenvalues, respectively. Now equation (3.7) becomes

$$\{I + \mu\Delta t\ (D_x^+ A^- + D_x^- A^+)\}\ \delta w + \Delta t\ R(w) = 0.$$

At the jth mesh point this is

$$\{I + \alpha(A_j^+ - A_j^-)\}\ \delta w_j + \alpha\ A_{j+1}^-\ \delta w_{j+1} - \alpha\ A_{j-1}^+\ \delta w_{j-1} + \Delta t R_j = 0$$

where

$$\alpha = \mu\ \frac{\Delta t}{\Delta x}$$

Set $\delta w_j^{(0)} = 0$. A two sweep symmetric Gauss-Seidel scheme is then

$$(1)\quad \left\{I + \alpha(A_j^+ - A_j^-)\right\} \delta w_j^{(1)} - \alpha\, A_{j-1}^+ \delta w_{j-1}^{(1)} + \Delta t R_j = 0$$

$$(2)\quad \left\{I + \alpha(A_j^+ - A_j^-)\right\} \delta w_j^{(2)} + \alpha\, A_{j+1}^- \delta w_{j+1}^{(2)} - \alpha\, A_{j-1}^+ \delta w_{j-1}^{(1)} + \Delta t R_j = 0$$

Subtracting (1) from (2) we find that

$$\left\{I + \alpha(A_j^+ - A_j^-)\right\} \delta w_j^{(2)} + \alpha\, A_{j+1}^- \delta w_{j+1}^{(2)} = \left\{I + \alpha(A_j^+ - A_j^-)\right\} \delta w_j^{(1)}$$

Define the lower triangular, upper triangular and diagonal operators L, U and D as

$$L \equiv I - \alpha\, A^- + \mu t\, D_x^-\, A^+$$

$$U \equiv I + \alpha\, A^+ + \mu t\, D_x^+\, A^-$$

$$D \equiv I + \alpha(A^+ - A^-)$$

It follows that the scheme can be written as

$$L\, D^{-1}\, U\, \delta w = -\, \Delta t\, R(w)$$

Commonly the iteration is terminated after one double sweep. The scheme is then a variation of an LU implicit scheme.

Some of these interconnections are illustrated in Figure 7. Schemes in three main classes appear to be the most appealing:

1) Varations of multi-stage time stepping, including the application of a Jacobi iterative method to the implicit scheme, (indicated by a single asterisk).

2) Variations of LU decomposition, including the application of a Gauss-Seidel iterative method to the implicit scheme (indicated by a double asterisk).

3) Alternating direction schemes, including schemes in which an LU decomposition is separately used in each coordinate direction (indicated by a triple asterisk).

The optimal choice may finally depend on the computer architecture. One might anticipate that the Gauss-Seidel method of iteration could yield a faster rate of convergence than a Jacobi method, and it appears to be a particularly natural choice in conjunction with a flux split scheme which yields diagonal dominance. This class of schemes, however, restricts the use of vector or parallel processing. Multistage time stepping, or Jacobi iteration of the implicit scheme, allow maximal use of vector or parallel processing. The alternating direction formulation removes any restriction on the time step (at least in the two dimensional case), while permitting vectorization along coordinate lines. The ADI-LU scheme is an interesting compromise.

5. Acceleration Methods: Multigrid Technique

Clearly one can anticipate more rapid convergence to a steady state as the time step is increased. Accordingly, the rate of convergence of an explicit scheme can generally be substantially improved by using a variable time step close to the local stability limit throughout the flow field. Assuming that the mesh cells are clustered near the body and expand as one moves away from the body, this effectively increases the rate at which disturbances are propagated through the outer part of the mesh. A similar strategy also pays with implicit schemes. In this case the terms in Δt^2 or Δt^3 resulting from factorization become dominant if Δt is too large, and the optimum rate of convergence is typically realized with a time step corresponding to a Courant number of the order of 10.

Radical further improvements in the convergence rate can be realized by the multigrid time stepping technique. The concept of acceleration by the introduction of multiple grids was first proposed by Federenko [46]. There is by now a fairly well developed theory of multigrid methods for elliptic equations [47-48], based on the concept of the updating scheme acting as a smoothing operator on each grid. This theory does not hold for hyperbolic systems. Nevertheless, it seems that it ought to be possible to accelerate the evolution of a hyperbolic system to a steady state by using large time steps on coarse grids so that disturbances will be more rapidly expelled through the outer boundary. Several multigrid time stepping schemes designed to take advantage of this effect have been proposed [49-56]. In each of them some of the task of tracking the evolution of the system is transferred to a sequence of coarser grids. Aside from helping

to establish equilibrium more rapidly, this has the advantage that the computational effort per time step is reduced on a coarser mesh. The interpolation of the corrections back to the fine grid will introduce errors, however, which cannot be rapidly expelled from the fine grid, and ought to be locally damped if a fast rate of convergence is to be attained. Thus it remains important that the driving scheme should have the property of rapidly damping out high frequency modes.

A multigrid time stepping scheme may be organized as follows. Suppose that successively coarser auxiliary grids are introduced, with the grids numbered from 1 to m, where grid 1 is the original mesh. Then after one or more time steps on grid 1 one passes to grid 2. Again, after one or more steps one passes to grid 3, and so on until grid m is reached. For $k > 1$, the evolution on grid k is driven by a weighted average of the residuals calculated on grid k-1, so that each mesh simulates the evolution that would have occurred on the next finer mesh. When the coarsest grid has been reached, changes in the solution calculated on each mesh are consecutively interpolated back to the next finer mesh. Time steps may also be included between the interpolation steps on the way back up to grid 1. In practice it has been found that an effective multigrid strategy is to use a simple saw tooth cycle, with one time step on each grid on the way down to the coarsest grid, and no Euler calculations between the interpolation steps on the way up.

In general one can conceive of a multigrid scheme using a sequence of independently generated coarser meshes which are not associated with each other in any structured way. Here attention will be restricted to the case in which

coarser meshes are generated by eliminating alternate points in each coordinate direction. Accordingly, each cell on grid k coincides either exactly or approximately with a group of four cells on grid k-1 in the two dimensional case, or eight cells in the three dimensional case. This allows the formulation of simple rules for the transfer of data between grids.

In order to give a precise description of the multigrid scheme it is convenient to use subscripts to indicate the grid. Several transfer operations need to be defined. First the solution vector on grid k must be initialized as

$$w_k^{(0)} = T_{k,k-1}\, w_{k-1}$$

where w_{k-1} is the current value on grid k-1, and $T_{k,k-1}$ is a transfer operator. Next it is necessary to transfer a residual forcing function such that the solution on grid k is driven by the residuals calculated on grid k-1. This can be accomplished by setting

$$P_k = Q_{k,k-1}\, R_{k-1}(w_{k-1}) - R_k(w_k^{(0)})$$

where $Q_{k,k-1}$ is another transfer operator. Then $R_k(w_k)$ is replaced by $R_k(w_k) + P_k$ in the time stepping scheme. For example, the multi-stage scheme defined by equation (3.3) is reformulated as

$$w_k^{(1)} = w_k^{(0)} - \alpha_1 \Delta t_k \,(R_k^{(0)} + P_k)$$

$$\cdots$$

$$w_k^{(q+1)} = w_k^{(0)} - \alpha_{q+1}\, \Delta t_k \,(R_k^{(q)} + P_k)$$

$$\cdots$$

The result $w_k^{(m)}$ then provides the initial data for grid k+1. Finally, the accumulated correction on grid k has to be transferred back to grid k-1. Let w_k^+ be the final value of w_k resulting from both the correction calculated in the time step on grid k and the correction transferred from grid k+1. Then one sets

$$w^{+}_{k-1} = w_{k-1} + I_{k-1,k}(w^{+}_{k} - w^{(0)}_{k})$$

where w_{k-1} is the solution on grid k-1 after the time step on grid k-1 and before the transfer from grid k, and $I_{k-1,k}$ is an interpolation operator.

In the case of a cell centered scheme the solution transfer operator $T_{k,k-1}$ is defined by the rule

$$T_{k,k-1}\ w_{k-1} = (\sum V_{k-1}\ w_{k-1})/V_k$$

where the sum is over the constituent cells on grid k-1, and V is the cell area or volume. This rule conserves mass, momentum and energy. The residual transferred to grid k is the sum of the residuals in the constituent cells

$$Q_{k,k-1}\ R_{k-1} = \sum R_{k-1}$$

The corrections are transferred up using either bilinear or trilinear interpolation for the operator $I_{k-1,k}$.

When the flow variables are stored at the cell vertices the solution transfer rule is simply to set $w^{(0)}_{k}$ to w_{k-1} at the coincident mesh point in grid k-1. The residual transfer rule is a weighted sum over the 9 nearest points in two dimensions, or the 27 nearest points in three dimensions. The corresponding transfer operator $Q_{k,k-1}$ can be expressed as a product of summation operators in the coordinate directions. Let μ_x denote an averaging operator in the x

direction:

$$(\mu_x R)_{i+1/2,j,k} = \frac{1}{2}(R_{i,j,k} + R_{i+1,j,k})$$

and

$$(\mu_x^2 R)_{i,j,k} = \frac{1}{4} R_{i-1,j,k} + \frac{1}{2} R_{i,j,k} + \frac{1}{4} R_{i+1,j,k}$$

Then in the three dimensional case

$$Q_{k,k-1} \equiv 8\ \mu_x^2\ \mu_y^2\ \mu_z^2$$

The interpolation operator $I_{k-1,k}$ transfers the corrections at coincident mesh points, and fills in the corrections at intermediate points by bilinear or trilinear interpolation.

To illustrate the power of the multigrid time stepping technique Figure 8 shows a solution of the inviscid Burger's equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0$$

The boundary conditions were set to allow a (nonunique) steady state solution, and adaptive dissipative terms were included to provide an oscillation free shock wave [39]. The calculation was performed by a three stage time stepping scheme in which the dissipative terms were evaluated twice. The fine mesh had 128 cells, and 5 levels were used in the multigrid scheme. Figure 8(a) shows the evolution of the solution of the first 10 steps, starting from the initial data at the bottom of the figure.

It can be seen that by the completion of the sixth step the solution is indistinguishable from the final steady solution, illustrated in Figure 8(b). In this case the symmetry of the initial data results in a shock wave with no interior points. The rate of convergence measured by the average value of

$\left|\frac{\partial u}{\partial}\right.$

Figure 9 shows the result of a multigrid calculation of three dimensional transonic flow past a swept wing.

The vertex formulation described by equations (2.5) was used for the discretization of the Euler equations. A five stage time stepping scheme was used in conjunction with a simple saw tooth multigrid cycle. Implicit residual averaging as defined by equation (4.5) was also used. The mesh was of C type in streamwise vertical planes, generated by the introduction of sheared parabolic coordinates. A mesh of this type contains a singularity where it folds beyond the wing tip, and the cells adjacent to the fold line are badly distorted. These cells, which have a very high aspect ratio and a triangular cross section, present a severe test of robustness of the multigrid scheme. Figure 9 shows a typical result for the well known ONERA M6 wing at a Mach number of .840 and an angle of attack of 3.06 degrees*. The mesh contained 96 cells in the chordwise direction, 16 cells in the direction normal to the wing, and 16 cells in the spanwise direction, and the calculation was performed in two stages. A result was first obtained on a 48x8x8 mesh using three levels in the multigrid scheme. This was then used to provide the initial state for the calculation on the 96x16x16 mesh in which four levels were used in the multigrid scheme. Table 1 shows the rate of convergence over 100 multigrid cycles on the 96x16x16 mesh,

*Calculated on a Cray 1 computer at Grumman: I am indebted to G. Volpe for his assistance in optimizing the computer program to run on the Cray and preparing the graphic display of the result.

measured by the average rate of change of density, together with the development of the lift and drag coefficients CL and CD. It can be seen that these are converged to four figures within 20 cycles. Table 2 shows the result of a similar calculation using a sequence of three meshes containing 32x8x8, 64x16x16 and 128x32x32 cells, respectively. Three levels were used in the multigrid scheme on the first mesh, four on the second, and five on the third. After 10 cycles on the 32x8x8 mesh, 10 cycles on the 64x16x16 mesh and 5 cycles on the 128x32x32 mesh, the calculated force coefficients were CL = .3145, and CD = .0167. These are barely different from the final converged values CL = .3144 and CD = .0164. The discretization errors, which may be estimated by comparing fully converged results on the sequence of three meshes, are, in fact, substantially larger than these differences. Thus convergence well within the discretization error can be obtained in 5-10 cycles.

Table 1

Calculation of the flow past the ONERA M6 wing at Mach .840, and 3.06° angle of attack on a 96x16x16 mesh.

Cycle	Average $d\rho/dt$	CL	CD
1	$.916\ 10^{-1}$		
10	$.158\ 10^{-2}$	.3110	.0205
20	$.243\ 10^{-3}$	.3118	.0203
30	$.245\ 10^{-4}$	.3118	.0203
40	$.353\ 10^{-5}$	.3118	.0203
50	$.528\ 10^{-6}$	.3118	.0203
60	$.772\ 10^{-7}$	.3118	.0203
70	$.124\ 10^{-8}$	.3118	.0203
80	$.241\ 10^{-9}$	.3118	.0203
90	$.363\ 10^{-9}$	.3118	.0203
100	$.528\ 10^{-10}$	.3118	.0203

Average reduction of $d\rho/dt$ per multigrid cycle: .807.

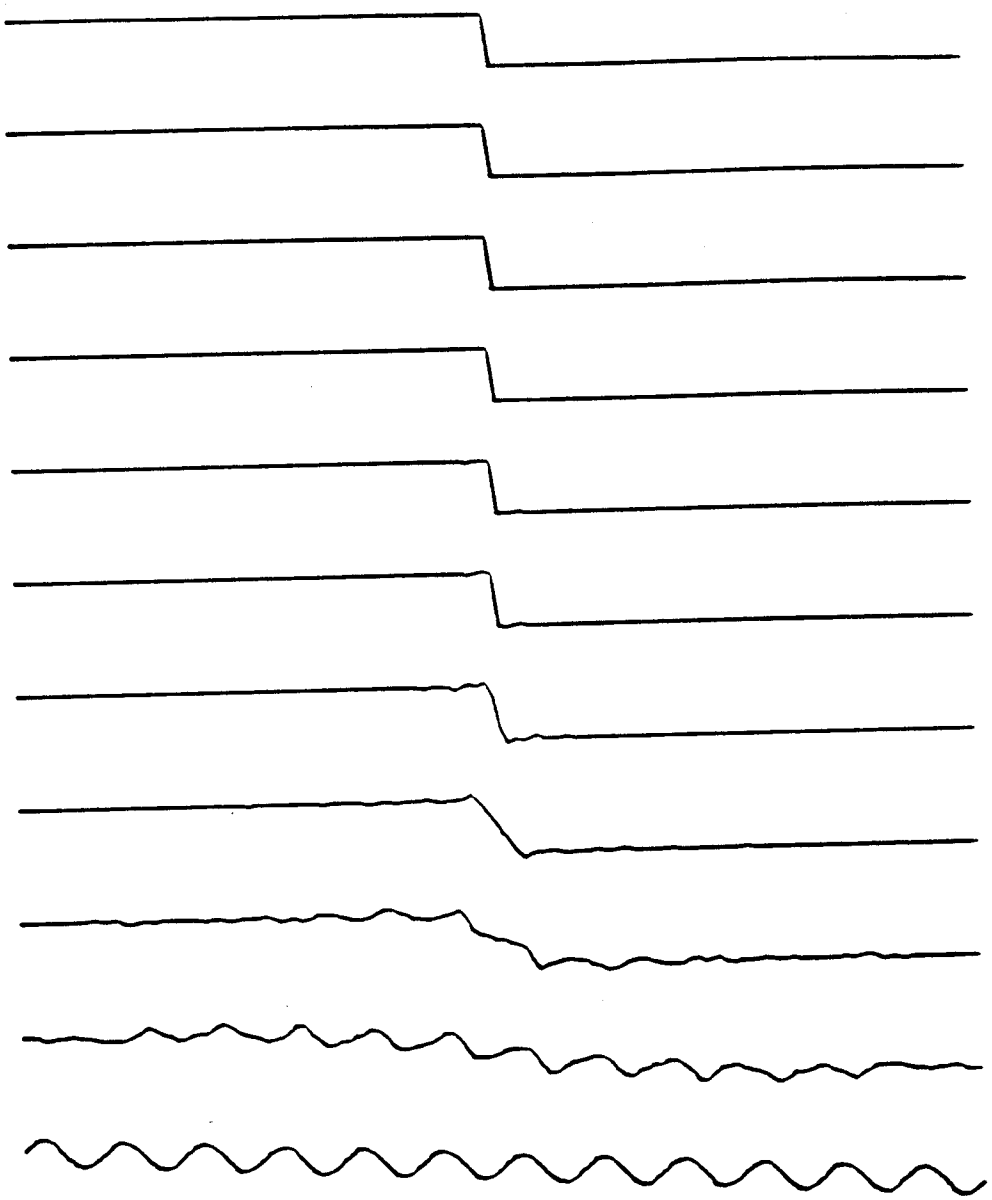


Figure 8(a)

Initial state and first 10 cycles in evolution of Burger's equation (reading upwards) Adaptive dissipation (scheme 1a) 128 cells 5 grids $\lambda = 2.0$

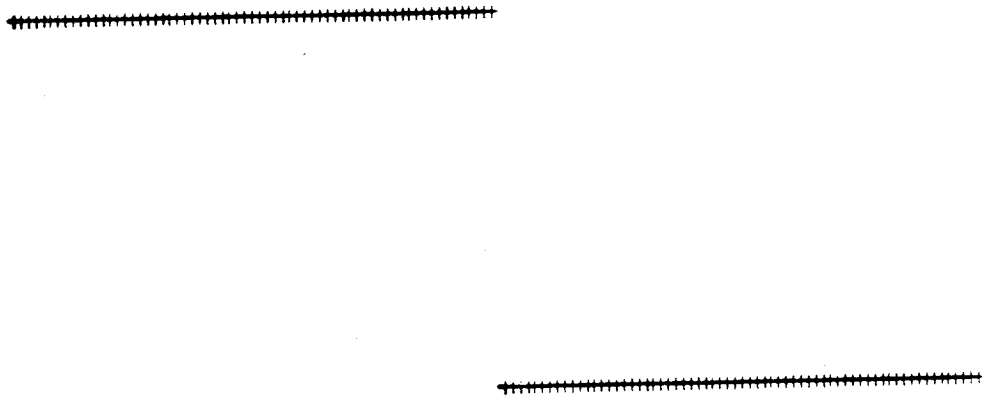


Figure 8(b)

Final State of Burger's equation
after 20 cycles of the multigrid scheme
Residual .5327 10-8

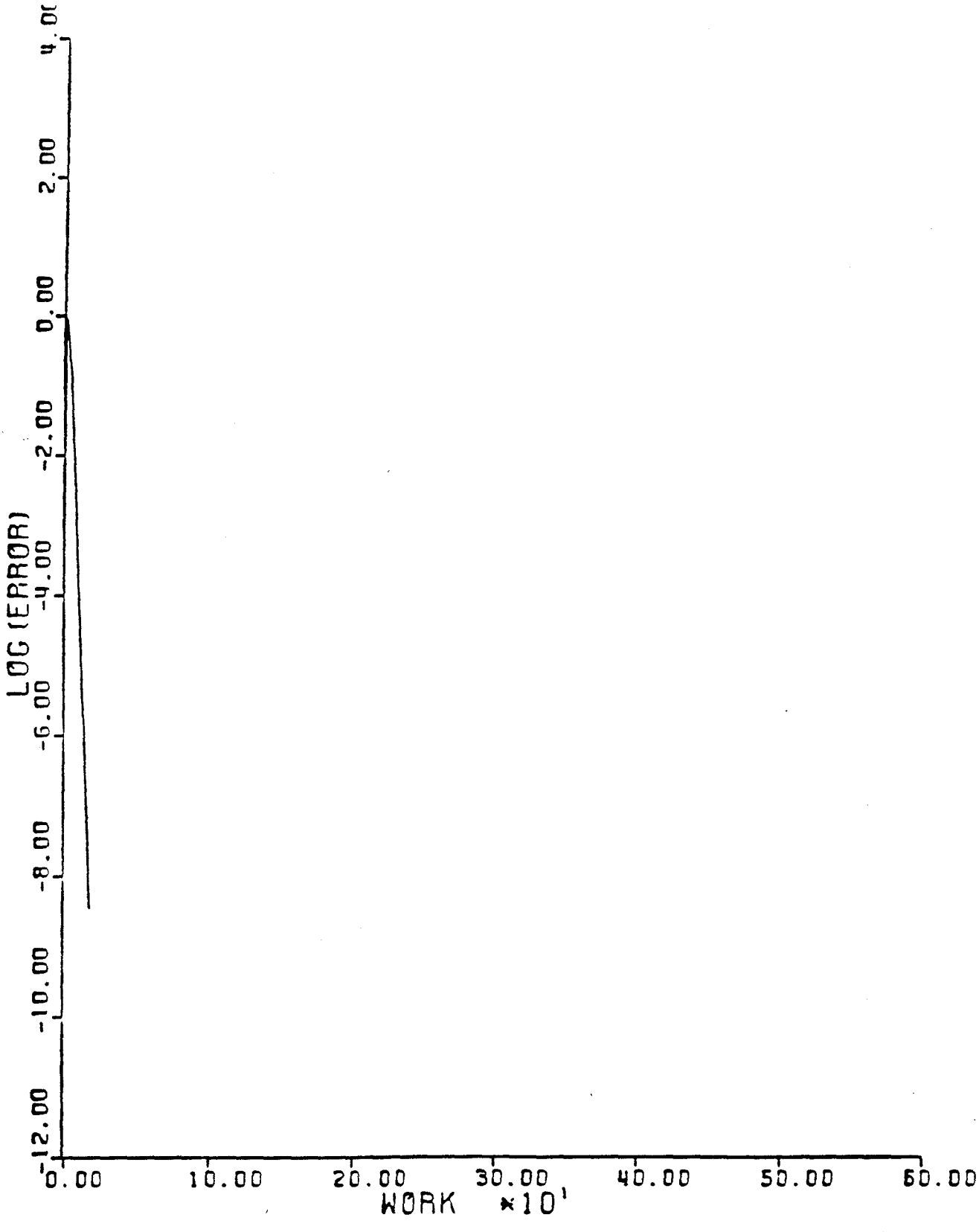


Figure 8(c)

Convergence history for Burger's equation
Adaptive Dissipation (scheme 1a)
128 cells 5 grids $\lambda = 2.0$
Mean rate of error reduction .3587 per cycle.

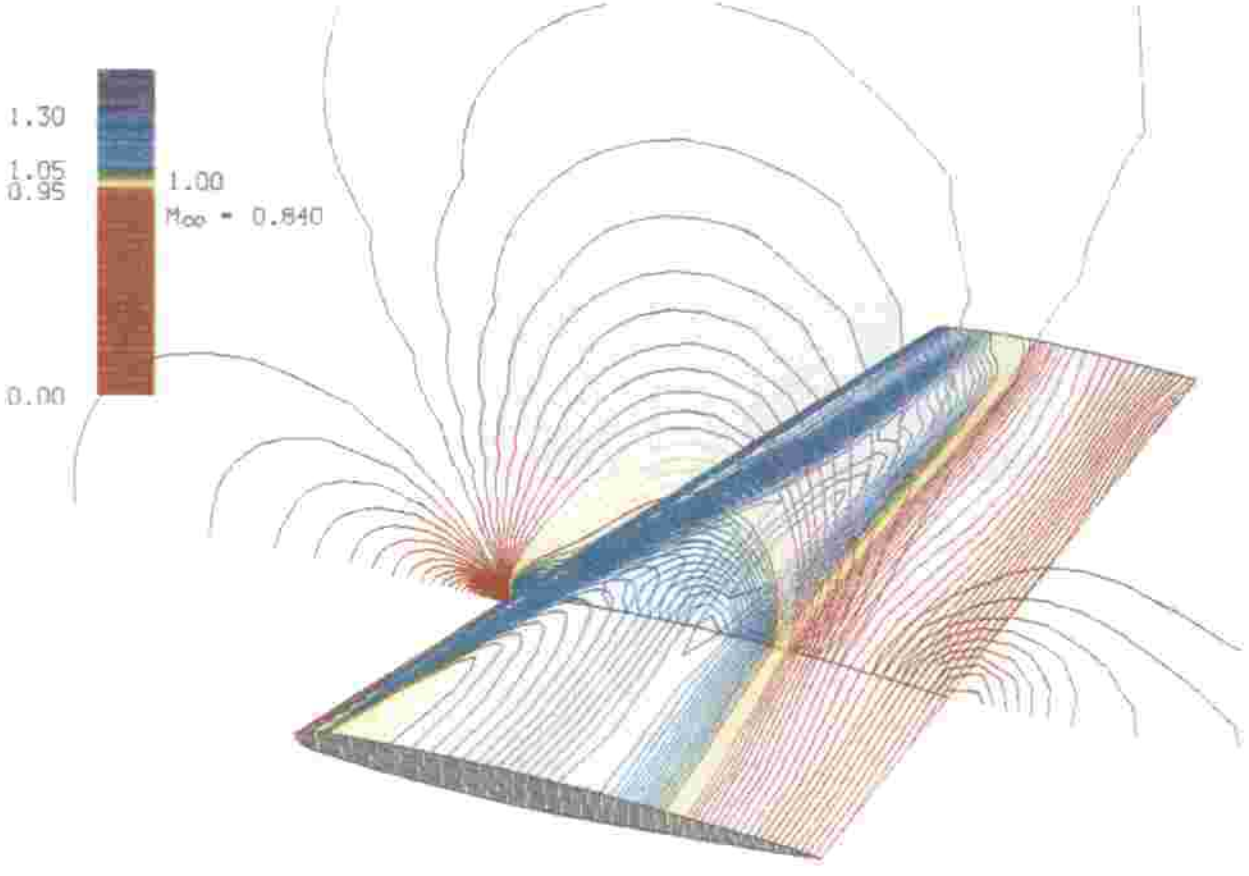


Figure 9

Constant pressure contours of flow over the ONERA M6 wing

6. Airplane Calculations

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 [13]: 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. Multiblock methods have been proposed as a method of generating meshes in very complex regions. In these the mesh is partitioned into smaller blocks so that the mesh generation problem in each individual block is simplified. The difficulty of defining the mesh, and ensuring contiguity of mesh lines at the various interfaces is still considerable. The constraints on each block can be relaxed by allowing the blocks to overlap, at the cost of the need for complex transfer procedures between the blocks.

Since an arbitrary set of points admits a triangulation, the problem can be simplified by separating the procedure for generating mesh points from the procedure for triangulating them. In this approach a cluster of mesh points surrounding the aircraft can be created in any convenient manner. An efficient method is to take the union of the points belonging to separately generated meshes around each component. No regularity is required in the initial point distribution, only that a reasonable point density is created corresponding to the anticipated variation in the flow field. 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 triangu-

lation 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: one procedure is to generate the Delaunay triangulation [57-61]. 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 than 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.

A strategy that has proved effective in practice 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. This approach to mesh generation has been implemented at Princeton University, and the finite element method outlined in Section 2 has been used to perform transonic flow calculations for complete aircraft [20].

Figure 10 shows 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 pressure 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. This 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 $.335\ 10^{2}$ to $.161\ 10^{-3}$ 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.

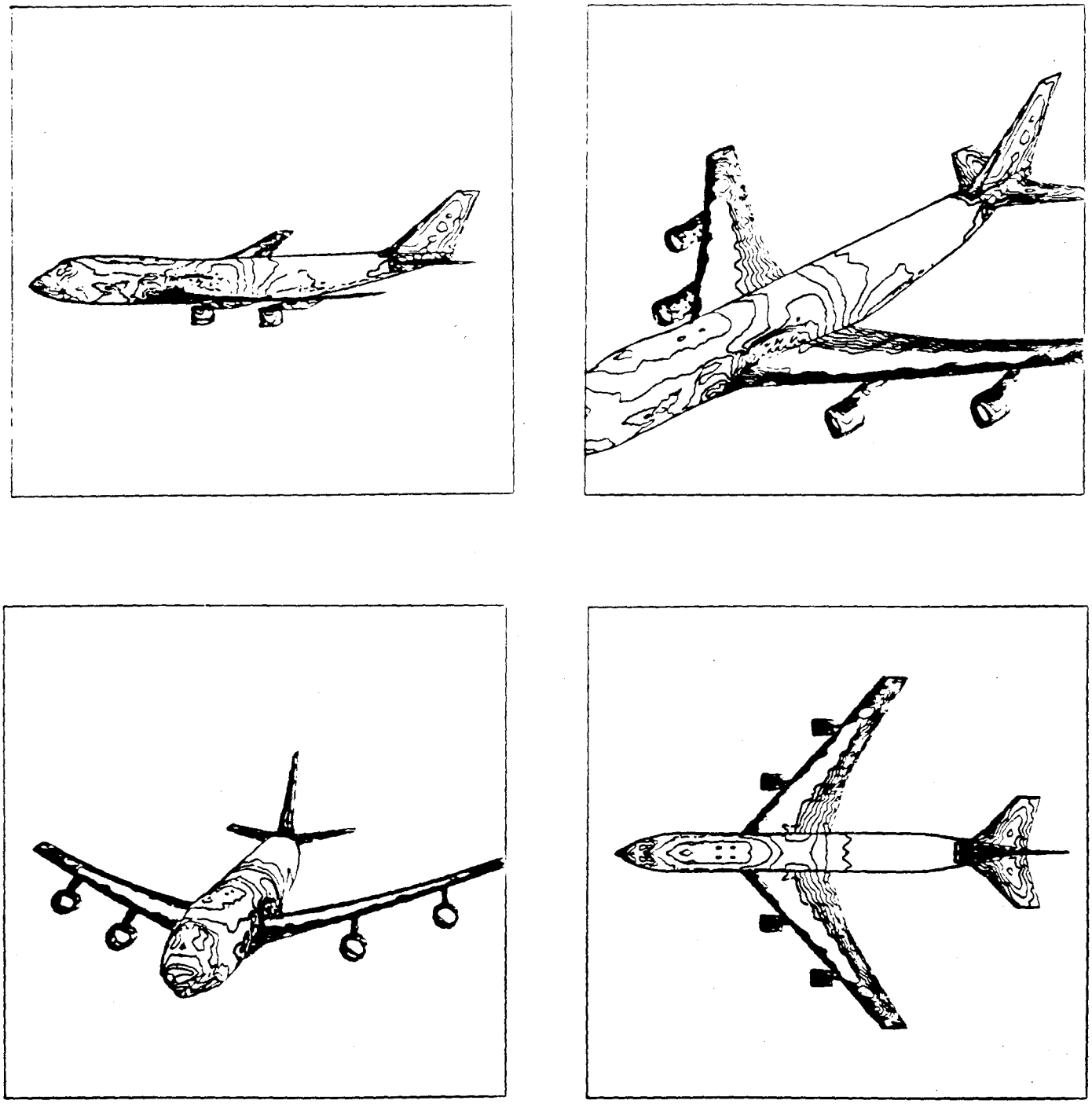


Figure 10

Surface Pressure Contours

7. Directions of Improvement

The results for the Boeing 747 clearly establish the feasability of performing transonic flow calculations for complete aircraft with the new generation of computers now becoming available. The present finite element method can benefit from a variety of improvements and extensions. 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 Lohner, Morgan and Peraire [62], and Holmes and Lamson [63].

(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 (2.7), 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.

8. Conclusion

Finite element methods on unstructured tetrahedral meshes, subject to some of the improvements outlined in Section 7, should provide a useful working tool which will allow engineers to assess the merits of proposed new designs. Parallel improvements in discretization and mesh generating methods for rectilinear meshes can also be anticipated. Thus we can look forward to an era in which transonic flow will routinely be computed for complete aircraft.

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. Bers, Lipman, "Mathematical Aspects of Subsonic and Transonic Gas Dynamics", Wiley, 1958.

4. Ferrari, C., and Tricomi, F.G., "Transonic Aerodynamics", Academic Press, 1968.

5. Morawetz, C.S., "On the Non-Existence of Continuous Transonic Flows Past Profiles", Comm. Pure. Appl. Math., Vol. 9, 1956, pp. 45-48.

6. Bauer, F., Garabedian, P., Korn, D., and Jameson, A., Supercritical Wing Sections II, Springer Verlag, 1975.

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

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

9. Jameson, A. and Caughey, D.A., Numerical Calculation of Transonic Flow Past a Swept Wing", New York University Report COO-3077-140, June 1977.

10. Jameson, A. 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.

11. 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.

12. 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.

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

14. 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.

15. 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.

16. Yakhot, V. and Orszag, S.A., "Renormalization Group Analysis of Turbulence, I. Basic Theory", Princeton University Applied and Computational Mechanics Report, 1986.

17. Giles, M., Drela, M., and Thompkins, W.T., "Newton Solution of Direct and Inverse Transonic Euler Equations", AIAA Paper 85-1530, Proc. AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 394-402.

18. Lax, P. and Wendroff, B., "Systems of Conservation Laws", Comm. Pure. Appl. Math., Vol. 13, 1960, pp. 217-237.

19. Jameson, A., and Mavriplis, D., "Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh", AIAA Paper 85-0435, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1985.

20. 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, January 1986.

21. Jameson, A., "Algorithms for Compressible Flow Calculations on Triangular and Tetrahedral Meshes", Princeton University Report MAE 1732, 1986.

22. Godunov, S. K., "A Difference Method for the Numerical Calculation of Discontinous Solutions of Hydrodynamic Equations", Mat. Sbornik, 47, 1959, pp. 271-306, translated as JPRS 7225 by U.S. Dept. of Commerce, 1960.

23. Boris, J. P., and Book, D. L., "Flux Corrected Transport. 1. SHASTA, A Fluid Transport Algorithm That Works", J. Comp. Phys. Vol. 11, 1973, pp. 38-69.

24. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. II, Monotonicity and Conservation Combined in a Second Order Scheme," J. Comp. Phys. Vol. 14, 1974, pp. 361-370.

25. Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gas Dynamics Equations with Applications to Finite Difference Methods," J. Comp. Phys., Vol. 40, 1981, pp. 263-293.

26. Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", J. Comp. Phys., Vol. 43, 1981, pp. 357-372.

27. Osher, S., and Solomon, F., "Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws", Math. Comp., Vol. 38, 1982, pp. 339-374.

28. Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws", J. Comp. Phys., Vol. 49, 1983, pp. 357-393.

29. Osher, S. and Chakravarthy, S., "High Resolution Schemes and the Entropy Condition", SIAM J. Num. Analysis, Vol. 21, 1984, pp. 955-984.

30. Sweby, P. K., "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws", SIAM J. Num. Analysis, Vol. 21, 1984, pp. 995-1011.

31. Anderson, B.K., Thomas, J.L., and Van Leer, B., "A Comparison of Flux Vector Splittings for the Euler Equations", AIAA Paper 85-0122, AIAA 23rd Aerospace Sciences Meeting, Reno, January, 1984.

32. Yee, H.C., "On Symmetric and Upwind TVD Schemes", Proc. 6th GAMM Conference on Numerical Methods in Fluid Mechanics, Gottingen, September 1985.

33. Jameson, A., "A Nonoscillatory Shock Capturing Scheme Using Flux Limited Dissipation", Lectures in Applied Mathematics, Vol. 22, Part 1, Large Scale Computations in Fluid Mechanics, edited by B. E. Engquist, S. Osher and R.C.J. Sommerville, AMS, 1985, pp. 345-370.

34. Lax, P.D., "Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves", SIAM Regional Series on Applied Mathematics, Vol. 11, 1973.

35. Jameson, A., and Lax, P.D., "Conditions for the Construction of Multi-Point Total Variation Diminishing Difference Schemes", Princeton University Report MAE 1650, April 1984.

36. Yoon, S. and Jameson, A., "An LU Implicit Scheme for High Speed Inlet Analysis", AIAA Paper 86-1520, AIAA/ASME/ASEE 22nd Joint Propulsion Conference, Huntsville, June 1986.

37. MacCormack, R.W., "The Effect of Viscosity in Hyper Velocity Impact Cratering", AIAA Paper 69-354, 1969.

38. Kinmark, I.P.E., "One Step Integration Methods with Large Stability Limits for Hyperbolic Partial Differential Equations", Advances in Computer Methods for Partial Differential Equations, V, edited by R. Vichnevetsky and R.S. Stepleman, IMACS, 1984, pp. 345-349.

39. Jameson, A., "Transonic Flow Calculations for Aircraft", Lecture Notes in Mathematics, Vol. 1127, Numerical Methods in Fluid Dynamics, edited by F. Brezzi, Springer Verlag, 1985, pp. 156-242.

40. Gourlay, A.R., and Mitchell, A.R., "A Stable Implicit Difference Scheme for Hyperbolic Systems in Two Space Variables", Numer. Math., Vol. 8, 1966, pp. 367-375.

41. Beam, R.W., and Warming, R.F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Form", J. Comp. Phys., Vol. 23, 1976, pp. 87-110.

42. Jameson, A., and Turkel E., "Implicit Schemes and LU Decompositions", Math. Comp. Vol. 37, 1981, pp. 385-397.

43. Obayashi, S., and Kuwakara, K., "LU Factorization of an Implicit Scheme for the Compressible Navier Stokes Equations", AIAA Paper 84-1670, AIAA 17th Fluid Dynamics and Plasma Dynamics Conference, Snowmass, June 1984.

44. Obayashi, S., Matsukima, K., Fujii, K., and Kuwakara, K., "Improvements in Efficiency and Reliability for Navier-Stokes Computations Using the LU-ADI Factorization Algorithm", AIAA Paper 86-0338, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

45. Chakravarthy, S.R., "Relaxation Methods for Unfactored Implicit Upwind Schemes", AIAA Paper 84-0165, AIAA 23rd Aerospace Sciences Meeting, Reno, January 1984.

46. Federenko, R.P., "The Speed of Convergence of One Iterative Process", USSR Comp. Math and Math Phys., Vol. 4, 1964, pp. 227-235.

47. Brandt, Achi, "Multi-Level Adaptive Solutions to Boundary Value Problems", Math. Comp., Vol. 31, 1977, pp. 333-390.

48. Hackbusch, W., "On the Multi-Grid Method Applied to Difference Equations", Computing, Vol. 20, 1978, pp. 291-306.

49. Ni, Ron Ho., "A Multiple Grid Scheme for Solving the Euler Equations", AIAA Journal, Vol. 20, 1982, pp. 1565-1571.

50. Jameson, A., "Solution of the Euler Equations by a Multigrid Method", Applied Math. and Computation, Vol. 13, 1983, pp. 327-356.

51. Hall, M.G., "Cell Vertex Multigrid Schemes for Solution of the Euler Equations", IMA Conference on Numerical Methods for Fluid Dynamics", Reading, April 1985.

52. Hemker, P.W., and Spekreijse, S.P., "Multigrid Solution of the Steady Euler Equations", Proc. Oberwolfach Meeting on Multigrid Methods, December 1984.

53. Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes", AIAA Paper 85-0293, AIAA 23rd Aerospace Sciences Meeting, Reno, Januaary, 1985.

54. Jameson, A., and Yoon, S., "LU Implicit Schemes with Multiple Grids for the Euler Equations", AIAA Paper 86-0105, AIAA 24th Aerospace Sciences Meeting, Reno, January, 1986.

55. Anderson, W.K., Thomas, J.L., and Whitfield, D.L., "Multigrid Acceleration of the Flux Split Euler Equations", AIAA Paper 86-0274, AIAA 24th Aerospace Sciences Meeting, Reno, January 1986.

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

57. Augenbaum, J.M., "A Lagrangian Method for the Shallow Water Equations Based on a Voronoi Mesh", J. Comp. Physics, Vol. 53, 1984, pp. 240-265.

58. McCartin, B., "Discretization of the Semiconductor Device Equations", in New Problems and New Solutions for Device and Process Modeling, Bode Press, 1985, pp. 72-82.

59. Watson, D.F., "Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes", The Computer Journal, Vol. 24, 1981, p.167-172.

60. Bowyer, A., "Computing Dirichlet Tessellations", The Computer Journal, Vol. 24, 1981, pp. 162-166.

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

62. 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.

63. 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.