COMPUTATIONAL TRANSONICS

In honor of the sixtieth birthday
of Paul Garabedian

By

Antony Jameson
Princeton University
Princeton, NJ 08544

Communications in Pure and Applied Mathematics,
Vol. 41, 1988, pp. 507−549

1 Introduction

This paper is written to commemorate the sixtieth birthday of Paul Garabedian. While Paul has made broad ranging contributions in mathematics, fluid dynamics and plasma physics, his work on computational aerodynamics had a very forceful impact at a critical juncture in the development of this subject. I believe that this work should be regarded as being very much in the tradition of classical applied aerodynamics. Airplanes fly by generating a well organized and carefully controlled flow over the lifting surfaces: consequently useful predictions do not necessarily require solution of full Navier Stokes equations, and can be accomplished with simplified mathematical models. The task of the applied mathematician is to identify a mathematical model which is relevant to the problem at hand and also amenable to solution. Viscous effects on full scale aircraft are essentially limited to a thin boundary layer which forms along the surface, and very little error is incurred by ignoring viscosity outside the boundary layer. Major advances were achieved in the course of the last century using the concept of an ideal fluid, both inviscid and incompressible, and this was the basis of a successful theory of airfoils and wings.

As flight speeds increased to the point that the flow was dominated by compressibility effects, the focus moved to transonic flow, and it was in this field that Paul garabedian's contributions were particularly significant. Transonic flow is important because it covers an efficient operating regime for long range aircraft. To a first approximation cruising efficiency is proportional to

$$\frac{ML}{D}$$

where M is the Mach number (speed divided by the speed of sound), L is the lift, and D is the drag. As long as the speed is well below the speed of sound, the lift to drag ratio does

not vary much with speed, so it pays to increase the speed until the effects of compressibility start to cause a radical change in the flow. This occurs when embedded pockets of supersonic flow appear, generally terminating in shock waves. A typical transonic flow pattern over a wing is illustrated in Figure 1. As the Mach number is increased the shock waves become strong enough to cause a sharp increase in drag, and finally the pressure rise through the shock waves becomes so large that the boundary layer separates. An unsteady buffeting flow may then develop, leading to violent vibrations which cannot be tolerated even by pilots of military aircraft. The most efficient cruising speed is usually in the transonic regime, just as the onset of drag rise, and the prediction of aerodynamic properties in steady transonic flow has been a primary challenge for applied aerodynamics. In the period from 1950−1970 a qualitative understanding of transonic aerodynamics was achieved by analytical methods, including the notable result of C. Morawetz that a shock free transonic flow solution must be an isolated point [1]. Useful quantitative predictions remained beyond the reach of analytical methods.

Paul Garabedian was one of the first to recognize that computers which had become available in the sixties had made possible the accurate and reliable numerical prediction of transonic flows, and that this could lead to significant improvements in the design of airplanes. He also went on to tackle some of the primary problems that needed to be solved to make this vision a reality.

It can be recognized that in order for a computational method to have an impact on design, it needs to meet several primary requirements

1) The mathematical model must provide an appropriate representation of the significant features of the flow.

2) The numerical method must be robust, not liable to fail when parameters are varied.

3) It should be possible to treat flows past useful configurations (at least a three dimensional wing).

4) The calculations should offer reasonable accuracy without incurring unreasonable costs.

Beyond these factors, Paul Garabedian also understood that the capability to analyze the flow past a given configuration was only a partial solution to the problem: the designer also needs guidance on how to produce a better shape. From the outset, therefore, he tackled both the analysis and the design problem.

The title of this paper is taken from a phrase which I first heard coined by Paul. I had the good fortune to collaborate with Paul on this subject in the period 1972−1980, after he hired me from the Grumman Aerospace Corporation. Before I arrived Paul and his student David Korn had already developed a method of designing shock free transonic airfoils based on the use of complex characteristics in the hodograph plane [2]. He was also in touch with R.T. Whitcomb, who had introduced a new class of supercritical airfoils which delayed the onset of drag rise [3]. The first Garabedian and Korn shock free lifting airfoils looked rather like the Whitcomb airfoils, with less abrupt variations of curvature. The design method was limited to the production of a single point design, valid for one Mach number and angle of attack. It provided no information about the flow in off-region conditions, since a change in the Mach number would lead to a change in the shock free shape, if one could be found. Therefore, Garabedian and Korn turned their attention to the problem of analyzing the flow past a given profile. Since the intent was to treat inviscid flows with quite weak shock waves, they considered it a sufficient approximation to ignore vorticity associated with entropy production by shock waves, and to assume the existence of a velocity potential. The

equation describing the flow can then be expressed in the quasilinear form as

$$\left(c^2 - u^2\right)\phi_{xx} - 2uv\phi_{xy} + \left(c^2 - v^2\right)\phi_{yy} \tag{1.1}$$

where u and v are Cartesian velocity components

$$u = \phi_x, \quad v = \phi_y \tag{1.2}$$

and the speed of sound c can be determined from the steady state energy equation

$$\frac{c^2}{\gamma - 1} + \frac{1}{2}q^2 = \text{constant} \tag{1.3}$$

where γ is the ratio of specific heats and q is the speed $\sqrt{u^2 + v^2}$. Equation (1.1) is to be solved subject to a Neumann boundary condition on the profile

$$\frac{\partial \phi}{\partial n} = 0 \tag{1.4}$$

where n is the normal direction, while ϕ is also required to satisfy a far field based on an asymptotic estimate of the solution. There is a change of type from elliptic to hyperbolic as the Mach number $M = q/c$ exceeds unity. In 1970 Murman and Cole published their celebrated scheme for the transonic small disturbance equation, in which a central difference approximation was used in the subsonic zone, and an upwind difference approximation in the supersonic zone [4]. Garabedian and Korn sought to generalize this scheme to the transonic potential flow equation (1.1). I was experimenting with similar ideas at Grumman. Subsequently, when I came to the Courant Institute we pooled out ideas to produce an improved computer program, and Frances Bauer added a procedure to correct the calculations from the effect of the boundary layer thickness. The resulting method proved extremely useful,

and Whitcomb used it extensively to improve his supercritical wing designs.

Paul Garabedian was also fully aware of the importance of treating real configurations and he asked me to develop a computer program for three dimensional flow. This program, which used the rotated difference scheme which I had devised as a generalization of the Murman−Cole scheme to provide a streamwise upwind bias in the supersonic zone [5], was first applied to yawed wings. These were being advocated by R.T. Jones as the most efficient solution for supersonic flight [6]. Subsequently, in collaboration with David Caughey, I modified the program to treat a swept wing, producing FLO22 [7], which remains in use to this day. Paul remained interested in the design problem, and he initiated a new approach in which the results of an analysis calculation were used to modify the shape as the calculation proceeded, with the aim of approaching a desired pressure distribution. This could be used in both two and three dimensional calculations, and led to a design method based on FLO22.

Many of the results of these investigations are summarized in a sequence of books [8−10]. Through them we gained a great deal of practical experience with the problems of scientific conputing, which involve an interplay between mathematical modeling, techniques of numerical approximation, mesh generation for the treatment of complex domains, and problems connected with the development of large scale computer programs with many interacting components. We also gained considerable insight into the mechanism of shock capturing and numerical shock structure [9, 11].

By 1980 computers had advanced in both speed and memory to the point where the cost of using the more complete mathematical model of inviscid flow provided by the Euler equations was no longer prohibitive. In our use of the potential flow approximation for transonic flow we had selected a model for wing design which allowed reasonable inexpensive calculations to be performed with the computers available in the seventies. As the shock

waves become stronger the model becomes less accurate, but by the time the errors are serious the shock wave boundary layer interaction may also be strong enough to cause separation, which could not be adequately represented without recourse to the full viscous flow equations. There are a variety of applications, however, where one needs to treat a rotational flow which is not necessarily dominated by viscous effects. Examples are the interaction of a wing with the flow induced by a prop-fan, of flow inside a compressor with rotating blades. In the analysis of complete configurations one must also allow for the interaction of vortices shed from lifting surfaces on other components. High speed aircraft often use extensions of the wing leading edge or a chined front fuselage to shed vortices which pass over the top of the wing, thus enhancing the lift. Apart from allowing a more accurate representaion of shock waves, the use of the full Euler equations of inviscid flow allows vortices to be captured like shock waves, without any need to track them explicitly. Since 1980 my own work has been concentrated mainly on developing methods for determining steady state solutions of the Euler equations, and the body of this paper is devoted to a discussion of this subject. I regard this work as a natural extension of our previous investigations based on the potential flow equation.

The development of computers capable of concurrent calculation through the use of vector and parallel processing elements has added a new factor beyond the criteria for an effective computational method outlined on page 3. One must now ask whether the method can make effective use of concurrent processing. The iterative procedure for solving the potential flow equation (1.1) was devised by embedding the steady state equation in an artifitial time dependent equation [5] of the form

$$\alpha_1\phi_{xt} + \alpha_2\phi_{yt} + \alpha_3\phi_t = \left(c^2 - u^2\right)\phi_{xx} - 2uv\phi_{xy} + \left(c^2 - v^2\right)\phi_{yy}$$

The characteristic cone of this equation is tangent to the $x-y$ plane, with the result that the

iterative scheme can only be properly consistent with the region of dependence defined by the cone if the values of the potential are updated sequentially, marching in appropriate directions in the $x - y$ plane. This inhibits concurrent processing. The requirement of sequential updating can be avoided by introducing a term containing ϕ_{tt}, and using a scheme with three time levels [12]. In the case of the Euler equations the initial boundary value problem for the real time dependent equations provides a natural framework for the development of iterative schemes which allow concurrent processing. This approach can be effective provided that the configuration is such that it generates a unique steady flow, independent of the initial condition. The design of far field boundary conditions to reduce spurious wave reflections is a source of difficulty, which can be ameliorated, however, by moving the outer boundary of the computational domain very far out.

The methods described in this paper attempt to meet both criteria (3) and (4) stated on page 2. By treating the conservation laws directly in integral form it is possible to devise discrete approximations based on an arbitrary subdivision of the domain into polyhedral cells. One approach to mesh generation is the introduction of a curvilinear coordinate system, leading to a subdivision into distorted cubic cells. Such a structured mesh allows simple identification of neighbors through triple indices, but is subject to topological constraints which make it hard to treat complex configurations. A subdivision into an entirely unstructured mesh of tetrahedral cells provides more flexibility in the treatment of complex configurations, but incurs additional computational overheads because of the need for pointers to identify neighbors. As an example of the power of this approach, a calculation for a complete aircraft is presented in the last section.

The space discretization leads to a semi-discrete model in which the evolution of the flow is described by a set of ordinary differential equations. To reduce computational costs as

required by criterion (4) it is important to find ways of accelerating the convergence of this system to a steady state. For this purpose the time step is locally varied with the cell size so that it becomes very large as the mesh expands toward the outer boundary. In effect this corresponds to increasing the signal propagation speed in the far field so that disturbances always reach the outer boundary in a finite number of steps. Moreover, if one assumes that the optimal time step increases with the space interval, then one can anticipate a faster rate of convergence on a coarser grid. This motivates the concept of time stepping on multiple grids. The cells of the fine mesh can be amalgamated into larger cells which form a coarser mesh. In each coarser mesh cell the conservation laws are then represented by summing the flux balances of its constituent fine mesh cells, with the result that the evolution on the coarse mesh is driven by the disequilibrium of the fine mesh equations. Finally, the corrections on the coarse mesh are interpolated back to the fine mesh. In the case of a structured mesh this process can be repeated through a sequence of meshes in each of which the mesh spacing is doubled from the previous. If the time step is also doubled each time the process passes to a coarser mesh, then a four level multigrid cycle consisting of one step on each mesh represents a total advance

$$\Delta t + 2\Delta t + 4\Delta t + 8\Delta t = 15\Delta t,$$

where Δt is the step on the fine mesh. Examples are presented to show that solutions converged to engineering accuracy can be obtained by this technique in as few as 10 to 30 steps. Similar methods can be applied to unstructured meshes, but they are much harder to program, and so far no attempt has been made to implement a multigrid method for a complete aircraft.

2 Space Discretization of the Euler Equations

It is convenient to derive a space discretization scheme for the Euler equations directly from the conservation laws for mass, momentum and energy expressed in integral form. 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

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

where γ is the ratio of specific heats. The Euler equations for flow of a compressible inviscid fluid can be written 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 $d\underline{S}$. 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 conservation of momentum in the x direction

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

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

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

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 2 for a two dimensional case.

Figures 2(a) and 2(b) show cell centered schemes on rectilinear and triangular meshes [13−16]. 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

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

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

Figure 2(c) and 2(d) show corresponding schemes on rectilinear and triangular meshes in which the flow variables are stored at the vertices [17−19]. 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 k^{th} cell in the control volume. The flux balance for a given cell is now approximated as

$$Q = \sum_l \underline{F}_l \cdot \underline{S}_l \tag{2.6}$$

where $\underline{S}_l$ is the directed area of the l^{th} face, and $\underline{F}_l$ 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 contribution 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,

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

in Figures 2(c) and 2(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. When a tetrahedral mesh is used in a three dimensional calculation, the use of a simple average of the three corner values of each triangular face

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

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)

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

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

$$\frac{\partial}{\partial t} \iiint_{\Omega} \phi w \, d\Omega = \iiint_{\Omega} \underline{F} \cdot \bigtriangledown \phi \, d\Omega - \iint_{\partial\Omega} \phi \underline{F} \cdot d\underline{S} \tag{2.7}$$

Suppose now that we take ϕ to be 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 $\bigtriangledown\phi$ 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

$$\phi_x = \frac{1}{V} \iiint \phi_x \, dx \, dy \, dz = \frac{1}{V} \sum_k S_{x_k} \bar{\phi}_k$$

where V is the cell volume, S_{x_k} and $\bar{\phi}_k$ are projected area of the k^{th} face in the x direction and the average value of ϕ on the k^{th} 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

$$\bigtriangledown \phi = -\frac{\underline{S}}{3V} \tag{2.8}$$

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 3, for example,

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

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

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

Since the control volume is closed, however,

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

Therefore the contribution of $\overline{\underline{F}}_0$ to $\overline{\underline{F}}_k$ can be discarded, leading to a sum over the faces multiplied by a constant. Thus if we write

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

for the average values of $\underline{F}$ on the face opposite vertex 0 we find that the right hand side of equation (2.7) can be replaced by

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

Taking w to be piecewise linear, the left hand side of equation (2.7) can be integrated explicitly. Canceling a factor of 1/4 on each side, the approximation finally reduces to

$$\sum_k V_k \left[\frac{2}{5}\frac{dw_0}{dt} + \frac{1}{5}\sum_{l=1}^{3}\frac{dw_{kl}}{dt}\right] + \sum_k \tilde{\underline{F}}_k \cdot \underline{S}_k = 0 \tag{2.9}$$

where the subscript l denotes the three exterior vertices of the k^{th} tetrahedron in the control

volume containing the node 0. Equation (2.5) is obtained by lumping the time derivatives in equation (2.9) at the center point, and leads to the same steady state solution. For an unsteady calculation equation (2.9) should be more accurate, but it is more expensive, since it calls for the solution of coupled equations at each time step.

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 interval 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.

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 alternatively 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 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 [20], a variety of dissipative and upwind schemes designed to have good shock capturing properties have been developed during the past decade [21−29]. Some of these are closely related to the upwind schemes which proved successful in the treatment of the potential flow equation. 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

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

of a solution of (3.1) does not increase, provided that any discontinuity appearing in the

solution satisfies an entropy condition [30]. The concept of total variation diminishing (TVD) difference schemes, introduced by Harten [26], 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

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

cannot increase. The general conditions for a multipoint one dimensional scheme to be TVD have been stated and proved by Jameson and Lax [31]. For a semi-discrete scheme expressed in the form

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

there conditions are

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

and

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

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

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

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

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 j^{th} cell is given by

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

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

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

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

$$a_{j+\frac{1}{2}} = \begin{cases} \frac{f_{j+1}-f_j}{v_{j+1}-v_j} & \text{if} \quad v_{j+1} \neq v_j \\ \frac{\partial f}{\partial v}\big|_{v=v_j} & \text{if} \quad v_{j+1} = v_j \end{cases} \tag{3.6}$$

More generally, if one sets

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

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

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

since one can write

$$\begin{aligned} h_{j+\frac{1}{2}} &= f_j + \frac{1}{2}\left(f_{j+1} - f_j\right) - \alpha_{j+\frac{1}{2}}\left(v_{j+1} - v_j\right) \\ &= f_j + \left(\frac{1}{2}a_{j+\frac{1}{2}} - \alpha_{j+\frac{1}{2}}\right)\left(v_{j+1} - v_j\right) \end{aligned}$$

and

$$\begin{aligned} h_{j-\frac{1}{2}} &= f_j - \frac{1}{2}\left(f_j - f_{j-1}\right) - \alpha_{j-\frac{1}{2}}\left(v_j - v_{j-1}\right) \\ &= f_j - \left(\frac{1}{2}a_{j-\frac{1}{2}} + \alpha_{j-\frac{1}{2}}\right)\left(v_j - v_{j-1}\right) \end{aligned}$$

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 initially 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 [26]. The use of antidiffusive terms and flux limiters to improve shock resolution can be traced to the work of Boris and Book [21]. The concept of the flux limiting was independently advanced by Van Leer [22].

A particular simple method of constructing a second order accurate TVD scheme is to introduce flux limiters directly into a higher-order dissipative term [32]. Define the numerical flux in equation (3.4) as

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

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

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

where

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

and $\alpha_{j+\frac{1}{2}}$ 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

$$\begin{aligned}\Delta x\frac{dv_j}{dt} =& -\frac{1}{2}a_{j+\frac{1}{2}}\Delta_{j+\frac{1}{2}} - \frac{1}{2}a_{j-\frac{1}{2}}\Delta_{j-\frac{1}{2}} \\ &- \alpha_{j+\frac{3}{2}}\Delta_{j+\frac{3}{2}} + 3\alpha_{j+\frac{1}{2}}\Delta_{j+\frac{1}{2}} \\ &- 3\alpha_{j-\frac{1}{2}}\Delta_{j-\frac{1}{2}} + \alpha_{j-\frac{3}{2}}\Delta_{j-\frac{3}{2}}\end{aligned}$$

where

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

This does not satisfy conditions (3.3) because of the coefficients of $\Delta_{j+\frac{3}{2}}$ and $\Delta_{j-\frac{3}{2}}$ 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

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

and define the function

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

Also let

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

Since ϕ satisfies the symmetry condition

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

it follows that

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

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

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

According to equation (3.12)

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

Therefore equation (3.4) now yields

$$\begin{aligned}
\Delta \frac{dv_i}{dt} = &- \frac{1}{2} a_{j+\frac{1}{2}} \Delta_{j+\frac{1}{2}} - \frac{1}{2} a_{j-\frac{1}{2}} \Delta_{j-\frac{1}{2}} \\
&+ \left(2 - \frac{\phi_{j+1}}{r_{j+1}} + \phi_j \right) \alpha_{j+\frac{1}{2}} \Delta_{j+\frac{1}{2}} \\
&- \left(2 - r_{j-1}\psi_{j-1} + \psi_j \right) \alpha_{j-\frac{1}{2}} \Delta_{j-\frac{1}{2}}
\end{aligned}$$

Since

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

and

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

it follows that conditions (3.3) are satisfied if condition (3.8) holds. In a region where the solution is smooth, say with e increasing, the forward flux limiter is active, so that (3.10) reduces to a one-sided formula of a second order.

The same concepts may be used in the treatment of a system of equations by treating each characteristic field separately. Let $A_{j+\frac{1}{2}}$ be a matrix with the property that the flux difference satisfies the relation

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

so that it is a numerical approximation to the Jacobian $\partial f/\partial w$. A matrix of this kind was first introduced by Roe, who also gave a procedure for its construction [24]. 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 $A_{j+\frac{1}{2}}$,

$$w_{j+1} - w_j = \sum \alpha_l \, r_l \tag{3.17}$$

Then

$$f_{j+1} - f_j = \sum \lambda_l \, \alpha_l \, r_l \tag{3.18}$$

and the desired dissipative tern can be constructed as

$$\sum \mu_l \, \alpha_l \, r_l \tag{3.19}$$

where

$$\mu_l \geq \frac{1}{2} \left|\lambda_l\right| \tag{3.20}$$

This method amounts to constructing separate dissipative terms for the characteristic variables defined by the eigenvectors of $A_{j+\frac{1}{2}}$. It is closely related to the concept of flux splitting first introduced by Steger and Warming [23], 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 baclwards.

Alternative methods of flux splitting, which lead to excellent shock capturing schemes have been propsed by Osher [25] and Van Leer [29].

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. It has been shown by Goodman and LeVeque that schemes which present growth of the two dimensional total variation are at best first order accurate [33]. An alternative is to devise schemes which prevent growth in the maximum norm. This does not guarantee the preservation of monotonicity, but in practice there are unlikely to be significant oscillations. 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 2. Assume that the evolution equation at the mesh point 0 depends on contribution from the nearest neighbors, numbered as in the figure. Suppose that this is expressed in the form

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

where the sum is over the neighbors. Then we require all the coefficients to be non-negative

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

This condition on the signs of the coefficients assures that a local maximum cannot increase, while a local minimum cannot decrease, and consequently it prevents growth in the maximum norm. If it is violated then there exist distributions of the values such that a local maximum

would 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 2(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

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

Defining the coefficient a_{k0} as

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

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

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

where the coefficient α_{k0} satisfy the condition

$$\alpha_{k0} \geq |a_{k0}| \tag{3.27}$$

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} sucj that

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

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

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

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

$$d_{i+\frac{1}{2},j,k} = R\left(\epsilon^{(2)} - \epsilon^{(4)}\delta_x^2\right)\left(\rho_{i+1,j,k} - \rho_{i,j,k}\right) \tag{3.29}$$

where δ_x^2 is the second difference operator, $\epsilon^{(2)}$ and $\epsilon^{(4)}$ 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 be made proportional to $1/\Delta t^*$. The coefficient $\epsilon^{(4)}$ 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 capturing is controlled by the coefficient $\epsilon^{(2)}$, which is made proportional to the normalized second difference of the pressure

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

in the adjacent cells. Schemes contructed 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 [34, 35]).

4 Time Stepping Schemes

The discretization 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 6 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. Explicit schemes, which might be considered, include linear multistep methods such as the leap frog and Adams−Bashforth schemes, and one step multi-stage methods such as the classical Runge−Kutta schemes. The one step multi-stage 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

$$\begin{aligned} w^{(0)} &= w^n \\ w^{(1)} &= w^{(0)} - \alpha_1 \Delta t R^{(0)} \\ &\dots \\ w^{(m-1)} &= w^{(0)} - \alpha_{m-1} \Delta R^{(m-2)} \\ w^{(m)} &= w^{(0)} - \Delta t R^{(m-1)} \\ w^{(n+1)} &= w^{(m)} \end{aligned} \tag{4.2}$$

The residual in the $q+1^{st}$ stage is evaluated as

$$R^{(q)} = \sum_{r=1}^{q} \beta_{qr} R(w^{(r)}) \tag{4.3}$$

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 [36]. 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

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

where

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

Blended multi-stage schemes of this type, which have been analyzed in reference 37, can be tailored to give a large stability intervals along both the imaginary and negative real axes.

The properties of multi-stage schemes can be further enhanced by residual averaging [15]. 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 j^{th} mesh point

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

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 nondissipative one dimensional case one needs

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

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

$$\left(1 - \epsilon_x \delta_x^2\right)\left(1 - \epsilon_y \delta_y^2\right)\left(1 - \epsilon_z \delta_z^2\right)\bar{R} = R \tag{4.5}$$

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 smooothing coefficients. Residual averaging can also be used on triangular meshes [16, 18]. The implicit equations are then solved by a Jacobi iterations.

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

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

can be linearized as

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

Equation (4.7) reduces to the Newton iteration if one sets $\mu = 1$ and lets $\Delta t \to \infty$. In a three dimendional case with an $N \times N \times N$ 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 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 [38], was given an elegant formulation for nonlinear problems by Beam and Warming [39]. In two dimensional case equation (3.7) is replaced by

$$\left(I + \mu\Delta t D_x A\right)\left(I + \mu\Delta t D_y B\right)\delta w + \Delta t R(w) = 0 \tag{4.8}$$

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 algorithm, which is amenable to vectorization by simultaneous solution of the tridiagonal equations along parallel coordinate lines, has been refined to a high

level of efficiency by Pulliam and Steger [40]. The idea of the LU decomposition method [41] 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

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

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. One can also combine an alternating direction scheme with LU decomposition in the difference coordinate direction [42].

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 Jocobi 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. The alternative of a symmetric Gauss−Seidel scheme has been successfully employed in several recent works [43, 44]. Consider the case of a flux split 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

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

At the j^{th} mesh point this is

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

where

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

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

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

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

Subtracting (1) from (2) we find that

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

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

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

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

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

It follows that the scheme can be written as

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

Commonly the iteration is terminated after one double sweep. The scheme is then a variation of an LU implicit scheme, and it has proven to be very effective [34, 35].

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

1) Variations 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 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 oprimal 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. Multi-stage 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 propogated 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 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 [45]. There is by now a fairly well developed theory of multigrid methods for elliptic equations [46−47], based on the concept of the updating scheme acting as a smoothing operator on each grid. This theory does not hold for hyberbolic 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 [48−52]. In each of them some of the task of tracking the evolution of the system is transferred to a sequence of coarser grids. This has two advantages. First, the computational effort per time step is reduced on a coarser mesh. Second, the use of larger control volumes on the coarser grids tracks the evolution on a larger scale, with the consequence that global equilibrium can be more rapidly attained. In the case of an explicit time stepping scheme, this manifests itself through the possibility of using successively larger time steps as the process passes to the coarser grids, without violating the stability bound.

In general one can conceive of a multigrid scheme using a sequence of independently generated coarse meshes which are not associated with each other in any structured way. It is convenient, however, to generate the coarser meshes by eliminating alternate points in each coordinate direction. 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 subscripts will be used 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. Thus, the multi-stage scheme defined by equation (4.2) is reformulated

$$\begin{aligned} w_k^{(1)} &= w_k^{(0)} - \alpha_1 \Delta t_k \left(R_k^{(0)} + P_k \right) \\ &\dots \\ w_k^{(q+1)} &= w_k^{(0)} - \alpha_{q+1} \Delta t_k \left(R_k^{(q)} + P_k \right) \\ &\dots \end{aligned}$$

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

$$w_{k-1}^+ = w_{k-1} + I_{k-1,k} \left(w_k^+ - w_k^{(0)} \right)$$

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 that the flow variables are stored at the mesh node, the solution transfer rule is simply to set $w_k^{(0)}$ 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:

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

and

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

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 correction at intermediate points by bilinear or trilinear interpolation. Corresponding transfer rules can easily be defined for cell centered schemes

The analysis of multigrid schemes is complicated by the nonuniformity of the process. If a mesh point is common to two meshes then corrections can be directly transferred from the coarse to the fine mesh. On the other hand the correction at a point of the fine mesh which is not contained in the coarse mesh has to be interpolated from the corrections at neighboring points. A useful insight can be gained by modeling the multigrid process as a combination of two processes. The first is a uniform process in which every mesh point is treated in the same way, and the second is a nonlinear filtering scheme which eliminates the

date from alternate points. As a simple example, consider a one dimensional model, in which each coarser mesh is produced by eliminating alternate points of the fine mesh. Figure 7(a) illustrates the data flow of a two level scheme in which grid 1 is the finer mesh and grid 2 is the coarser mesh. Suppose that the calculation is simulating an equation of the form

$$\frac{du_j}{dt} = R_j(u) \tag{5.1}$$

where u_j is the dependent variable at mesh point j of grid 1, and $R_j(u)$ is the residual. Here it is convenient to use bracketed superscripts to indicate the grid level, and to reserve the use of subscripts for the indication of the location of the mesh point in the fine grid. Suppose that the points $0, 2, 4 \ldots$ are common to both meshes, while the points $1, 3, 5 \ldots$ are eliminated in grid 2. A simple multigrid scheme can be described as follows. One grid 1 u_j is updated by a correction

$$\delta u_j^{(1)} = -\Delta t^{(1)} f\left(R_j(u)\right) \tag{5.2}$$

where the function f depends on the time stepping scheme. On grid 2 corrections are calculated as

$$\delta u_j^{(2)} = -\Delta t^{(2)} f(R_j^{(2)}), \qquad j = 1, 3, 5 \ldots \tag{5.3}$$

where the residual $R_j^{(2)}$ is calculated by accumulating the residuals at the nearest neighbors after first allowing for the correction introduced on grid 1. For example,

$$R_j^{(2)} = \epsilon R_{j-1}^{+} + 2\left(1-\epsilon\right) R_j^{+} + \epsilon R_{j+1}^{+} \tag{5.4}$$

where the degree of spreading is controlled by the parameter ϵ, and

$$R_j^{+} = R_j\left(u + \delta u^{(1)}\right) \tag{5.5}$$

Then on interpolating the corrections on grid 2 back to grid 1, the total correction of the complete multigrid scheme is

$$\begin{aligned} \delta u_j &= \delta u_j^{(1)} + \delta u_j^{(2)}, && j \quad \text{even} \\ \delta u_j &= \delta u_j^{(1)} + \tfrac{1}{2}\left(\delta u_{j-1}^{(2)} + \delta u_{j+1}^{(2)}\right), && j \quad \text{odd} \end{aligned}$$

This process can be broken down into two stages as illustrated in Figure 7(b). First the corrections $\delta u_j^{(2)}$ are calculated for all points of grid 1 by formulas $(5.3) - (5.5)$ for j both even and odd. In effect the two level process is now calculated uniformly on the original fine grid. In the second stage $\delta u_j^{(2)}$ is then replaced by

$$\begin{aligned} \delta \bar{u}_j^{(2)} &= \delta u_j^{(2)}, && j \quad \text{even} \\ \delta \bar{u}_j^{(2)} &= \tfrac{1}{2}\left(\delta u_{j-1}^{(2)} + \delta u_{j+1}^{(2)}\right), && j \quad \text{odd} \end{aligned}$$

This nonlinear filtering process eliminates the need to calculate $\delta u_j^{(2)}$ at the odd points, allowing these calculations to be shifted to a coarser grid. It introduces an additional error

$$\begin{aligned} e_j &= 0, && j \quad \text{even} \\ e_j &= \tfrac{1}{2}\left(\delta u_{j-1}^{(2)} - 2\delta u_j^{(2)} + \delta u_{j+1}^{(2)}\right), && j \quad \text{odd} \end{aligned}$$

Assuming the mesh to be uniform, this can be written as

$$e_j = \frac{1}{4}\left(\delta u_{j-1}^{(2)} - 2\delta u_j^{(2)} + \delta u_{j+1}^{(2)}\right)\left(1 - \cos\frac{\pi}{\Delta x} x_j\right) \tag{5.6}$$

where Δx is the mesh interval of the fine mesh, and $x_j = j\Delta x$ are its mesh points. Thus the filter introduces additional errors in the form of a carrier wave at the mesh frequency $\pi/\Delta x$ of the fine mesh, modulated by the second difference of the corrections $\delta u_j^{(2)}$ which would be calculated in the second stage of the uniform scheme.

If we make the usual assumptions of linearity and periodicity, the multilevel uniform scheme can be analyzed by the Fourier method. If the multilevel uniform scheme is unstable, we can anticipate that the corresponding multigrid scheme will be unsound. Because of the injection of additional errors at various mesh frequencies by the interpolation process of the multigrid scheme, a reasonable criterion is to require the multilevel uniform schemes to have a substantial stability margin at the mesh frequencies of all the meshes above the coarsest mesh in the sequence. The Fourier analysis of the multilevel scheme requires careful tracking of the transfer of residuals and corrections between the meshes. It is shown in reference [52] that ths amplification factor of the composite scheme can be expressed by a recursion formula.

With properly optimized coefficients the multi-stage time stepping scheme is a very efficient driver of the multigrid process. Also a W cycle of the type illustrated in Figure 8 proves to be a particularly effective strategy for managing the work split between the meshes. Some results are presented in Figure 9. Figure 9 shows a result for the RAE 2822 airfoil computed over an O mesh with 160 cells around the profile and 32 cells in the normal direction. This was obtained with a five stage time stepping scheme in which the dissipative terms were evaluated three times in each step. A cell centered formulation was used for the space discretization, with adaptive dissipation of the type defined by equations (3.28) and (3.29). The average residual measured by the rate of change of the density was reduced from .124 to $.219 \times 10^{-10}$ in 100 cycles. This corresponds to an average reduction of .797 per cycle. The solution after 10 cycles is also displayed, and it can be seen that the solution is virtually identical. The lift coefficient is 1.1258 after 10 cycles and 1.1256 after 100 cycles. The program also has an option to use flux difference splitting with flux limited dissipation. Figure 10 shows the result with this option. The shock wave is sharper, but the leading edge section peak is underestimated, with the result that a lower lift coefficient of 1.1155

is predicted. This seems to be a consequence of greater production of spacious entropy at the front stagnation point. The convergence rate is also slower, with a mean reduction of .9156 per cycle. Figure 11 shows a three dimensional calculation for a swept wing using a vertex scheme on a $144 \times 24 \times 24$ mesh. In this case the mean convergence rate over 100 cycles if .8079 and a fully converged result is obtained in 15 cycles. Computer times for these calculations are small enough in their use that an interactive design method could be contemplated. A two dimensional calculation with 10 cycles on a 160×32 mesh can be performed on a Cray in several seconds. A three dimensional calculation with 15 cycles on $96 \times 16 \times 16$ mesh requires about 25 seconds using one processor of a Cray XMP. The multigrid technique can also be used to accelerate calculations on unstructured meshes [53]. In this case the programming is much more complicated, and it pays to store the transfer operators to avoid the need for repeated searches to identify contiguous cells on different meshes.

6 Airplane Calculation

If computational methods are to be really useful to airplane designers, they must be able to treat extremely complex configurations, ultimately extending up to a complete aircraft. A major pacing item of the effort to attain this goal has been the problem of mesh generation. For simple wing body combinations it is possible to generate rectilinear meshes without too much difficulty [54]: 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 procedure is to use tetrahedral cells in an unstructured mesh which

can be adapted to conform to the complex surface of an aircraft. References [18] and [19] present a method based on such an approach. 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. 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 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: one procedure is to generate the Delaunay triangulation [55−58]. 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. The insertion of a new point within the Dalaunay sphere of one of these interior tetrahedra would cause a reconnection

penetrating the surface. Therefore, as new points are added, it is necessary to test for this eventuality, and whenever it occurs the reconnection is modified to preserve the integrity of the solid boundary. This approach to mesh generation has been combined with the finite element method outlined in Section 2 to perform flow calculations for complete aircraft [18, 19].

Figure 12 shows the result of a transonic flow calculation for a Boeing 747−200 flying at Mach .84 at an angle of attach 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 modeled as open tubes. The mesh contains 35370 points and 181952 tetrahedra. The calculation was performed at Cray Research on a Cray XMP 216. The mesh generation and triangulation took 1100 seconds. The flow solution was calculated with 400 time steps in a further 3300 seconds, giving a total computation time of about 1 1/4 hours. This was performed with 400 cycles of the three stage scheme. Implicit smoothing with a smoothing parameter $\epsilon = 1$ 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 2.42×10^2 to 5.13×10^{-3} after 400 cycles. Although the mesh is fairly coarse, the significnant features of the flow are evident, inluding 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 just under 12 million words of memory. 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.

7 Conclusion

Conputational methods are now an indispensable tool of applied aerodynamics. Advances in computing power and numerical algorithms have brought us to the point where compressible flow calculations are routinely performed by a veriety of well established methods. Problems associated with the capture of shock waves and contact discontinuities are clearly understood. While discretization procedures suitable for the treatment of complex configurations have been developed, the computing resources required to produce useful answers continue to test the limits of currently available hardware. Automatic local mesh refinement offers a promising avenue for economical improvement of the resolution of features of complex flows [53, 59]. There are widespread efforts underway to develop accurate and reliable methods of predicting viscous compressible flows. Adequate resolution of the boundary later requires the use of very fine meshes [60]. The flow in the boundary layer is generally turbulent, and advances in turbulence modeling will also be needed to provide satisfactory simulations of viscous transonic flows. Eventually the design problem, already addressed by Paul Garabedian for transonic airfoils and wings, ought to be restored to the position of prominence which it deserves. To extract the full benefits of computational flow simulations, automatic methods need to be devised for improving the aerodynamic configuation, subject to other design constraints such as structural weight and contained volume.

References

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

[2] Garabedian, P.R. and Korn, D.G., "Numerical Design of Transonic Airfoils", Proc. SYNSPADE 1970, edited by B. Hubbard, Academic Press, New York, 1971, pp. 253−271.

[3] Whitcomb, R.T., "Review of NASA Supercritical Airfoils", 9^{th} ICAS Congress, Haifa, Paper 74−100, 1974.

[4] Murman, E.M. and Cole, J.D., "Calculation of Plane Steady Transonic Flows", AIAA Journal, 9, 1971, pp. 114−121.

[5] Jameson, A., "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1", Comm. Pure. Appl. Math, 27, 1974, pp. 283−309.

[6] Jones, R.T., "Reduction of Wave Drag by Antisymmetric Arrangement of Wings and Bodies", AIAA Journal, 10, 1972, pp. 172−176.

[7] Jameson, A. and Caughey, D., "Numerical Calculation of the Transonic Flow Past a Swept Wing", ERDA Report, C00−3077−140, 1977.

[8] Bauer, F., Garabedian, P. and Korn, D., "A Theory of Supercritical Wing Sections, with Computer Examples", Springer, 1972.

[9] Bauer, F., Garabedian, P., Korn, D. and Jameson, A., "Super Critical Wing Sections II", Springer, 1975.

[10] Bauer, F., Garabedian, P. and Korn, D., "Super Critical Wing Sections III", Springer, 1977.

[11] Jameson, A., "Numerical Solution of the Nonlinear Partial Differential Equations of Mixed Type", Proc. SYNSPADE 1975, edited by B. Hubbard, Academic Press, 1976, pp. 275−320.

[12] Jameson, A. and Keller, J.D., "Preliminary Study of the Use of the STAR−100 Computer for Transonic Flow Calculations", AIAA Paper 78−12, January 1978.

[13] Jameson, A., "Transonic Aerofoil Calculations Using the Euler Equations", Proc. IMA Conference on Numerical Methods in Aeronautical Fluid Dynamics, Reading, 1981, edited by P.L. Roe, Academic Press, 1982, pp. 289−308.

[14] Jameson, A., Schmidt, W. and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes", AIAA Paper 81−1259, AIAA 14^{th} Fluid Dynamics and Plasma Dynamics Conference, Palo Alto, 1981.

[15] Jameson, A. and Baker, T.J., "Solution of the Euler Equations for Complex Configurations", Proc. AIAA 6^{th} Computational Fluid Dynamics Conference, Danvers, 1983, pp. 293−302.

[16] Jameson, A. and Mavriplis, D., "Finite Volume Solution of the Two−Dimensional Euler Equations on a Regular Triangular Mesh", AIAA Paper 85−0435, AIAA 23^{rd} Aerospace Sciences Meeting, Reno, January, 1985.

[17] Jameson, A., "A Vertex Based Multigrid Algorithm for Three Dimensional Compressible Flow Calculations", in Numerical Methods for Compressible Flow − Finite Difference, Element and Volume Techniques, edited by T.E. Tezduar and T.J.R. Hughes, ASME Publication AMD 78, 1986.

[18] Jameson, A., Baker, T.J. and Weatherill, N., "Calculation of Inviscid Transonic Flow over a Complete Aircraft", AIAA Paper 86−0103, AIAA 24^{th} Aerospace Sciences Meeting, Reno, January 1986.

[19] Jameson, A. and Baker, T.J., "Improvements to the Aircraft Euler Method", AIAA Paper 87−0452, AIAA 25^{th} Aerospace Sciences Meeting, Reno, January 1987.

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

[21] 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.

[22] 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.

[23] 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.

[24] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", J. Comp. Phys., Vol. 43, 1981, pp. 357−372.

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

[26] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws", J. Comp. Phys., Vol. 49, 1983, pp. 357−393.

[27] Osher, S. and Chakravarthy, S., "High Resolution Schemes and the Entropy Condition", SIAM J. Num. Analysis, Vol. 21, 1984, pp. 955−984.

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

[29] Anderson, B.K., Thomas, J.L. and Van Leer, B., "A Comparison of Flux Vector Splitting for the Euler Equations", AIAA Paper 85−0122, AIAA 23^{rd} Aerospace Sciences Meeting, Reno, January, 1984.

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

[31] 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.

[32] 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. Commerville, AMS, 1985, pp.345−370.

[33] Goodman, J.B. and LeVeque, R.J., "On the Accuracy of Stable Schemes for 2−D Scalar Conservation Laws", Math. Comp., 45, 1985, pp.15−21.

[34] Yoon, S. and Jameson A., "An LU Implicit Scheme for High Speed Inlet Analysis", AIAA Paper 86−1520, AIAA/ASME/ASEE 22^{nd} Joint Propulsion Conference, Huntsville, June 1986.

[35] Jameson, A. and Reiger, H., "Solution of Steady Three−Dimensional Compressible Euler and Navier−Stokes Equations by an Implicit LU Scheme", AIAA Paper 88−0619, AIAA 26^{th} Aerospace Sciences Meeting, Reno, January 1988.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] Pulliam, T.H. and Steger, J.L., "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms", AIAA 85−0360, AIAA 23^{rd} Aerospace Sciences Meeting, Reno, January 1985.

[41] Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompositions", Math. Comp. Vol. 37, 1981, pp. 385−397.

[42] Obayashi, S. and Kuwakara, K., "LU Factorization of an implicit Scheme for the Compressible Navier Stokes Equations", AIAA Paper 84−1670, AIAA 17^{th} Fluid Dynamics and Plasma Dynamics Conference, Snowmass, June 1984.

[43] MacCormack, R.W., "Current Status of Numerical Solutions of the Navier−Stokes Equations", AIAA Paper 85−0032, AIAA 23^{rd} Aerospace Sciences Meeting, Reno, January 1985.

[44] Chakravarthy, S.R., "Relaxation Methods for Unfactored Implicit Upwind Schemes", AIAA Paper 84−0165, AIAA 22^{rd} Aerospace Sciences Meeting, Reno, January 1984.

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

[46] Brandt, A., "Multi−Level Adaptive Solutions to Boundary Value Problems", Math. Comp., Vol. 31, 1977, pp. 333−390.

[47] Hackbusch, W., "On the Multi−Grid Method Applied to Difference Equations", Computing, Vol. 20, 1978, pp. 291−306.

[48] Ni, R.H., "A Multiple Grid Scheme for Solving the Euler Equations", AIAA Journal, Vol. 20, 1982, pp. 1565−1571.

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

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

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

[52] Jameson, A., "A Multigrid Algorithm for Compressible Flow Calculations", Second European Conference on Multigrid Methods, Cologne, October 1985, Princeton University Report MAE 1743.

[53] Jameson, A. and Mavriplis, D., "Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes", 3^{rd} Copper Mountain Conference on Multigrid Methods, Copper Mountain, April 1987.

[54] Baker, T.J., "Mesh generation by a Sequence of Transformations", Applied numerical Mathematics, 2, 1985, pp. 515−528.

[55] Delaunay, B., "Sur la Sphere vide", Bull. Acad. Science USSR VII: Class Scil. Mat. Nat., 1934, pp. 793−800.

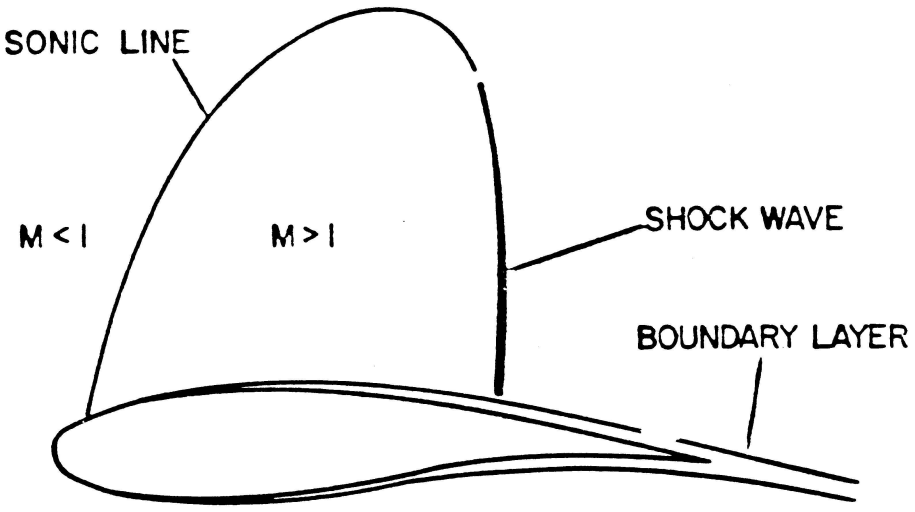
[56] Voronoi, G., "Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques. Deuxieme Memoire: Recherches Sur les Parallelloedres Primitifs", J. Reine Angew. Math., 134, 1908, pp. 198−287.

[57] Bowyer, A., "A Computing Dirichlet Tessellations", The Computer Journal, Vol. 24, 1981, pp. 162−166.

[58] Baker, T.J., "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets", AIAA Paper 87−1124−CP, AIAA 8^{th} Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1987.

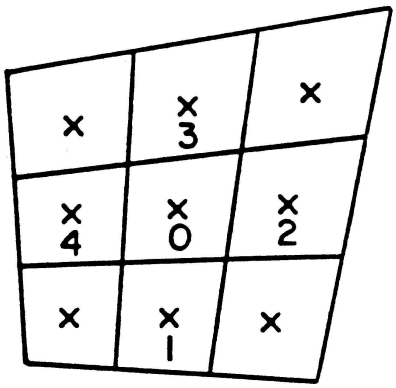
[59] Berger, M. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations", AIAA Journal, 23, 1985, pp. 561−568.

[60] Martinelli, L. and Jameson, A., "Validation of a Multigrid Method for the Reynolds Averaged Equations", AIAA Paper 88−0414, AIAA 26^{th} Aerospace Sciences Meeting, Reno, January 1988.



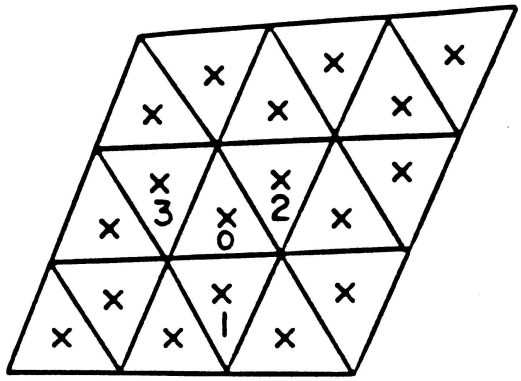
TRANSONIC FLOW PAST AN AIRFOIL

Figure 1



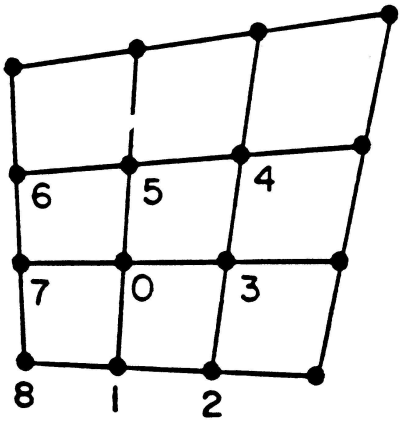
(a)

CELL CENTERED RECTILINEAR



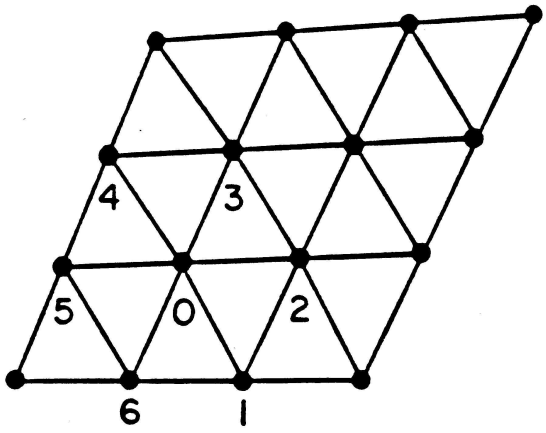
(b)

CELL CENTERED TRIANGULAR



(c)

VERTEX RECTILINEAR



(d)

VERTEX TRIANGULAR

Figure 2

Alternative discretization schemes

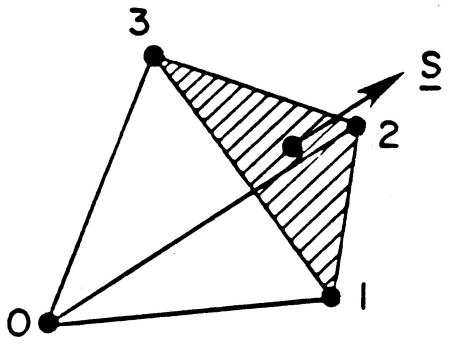


Figure 3

One tetrahedron of the control volume centered at node 0.

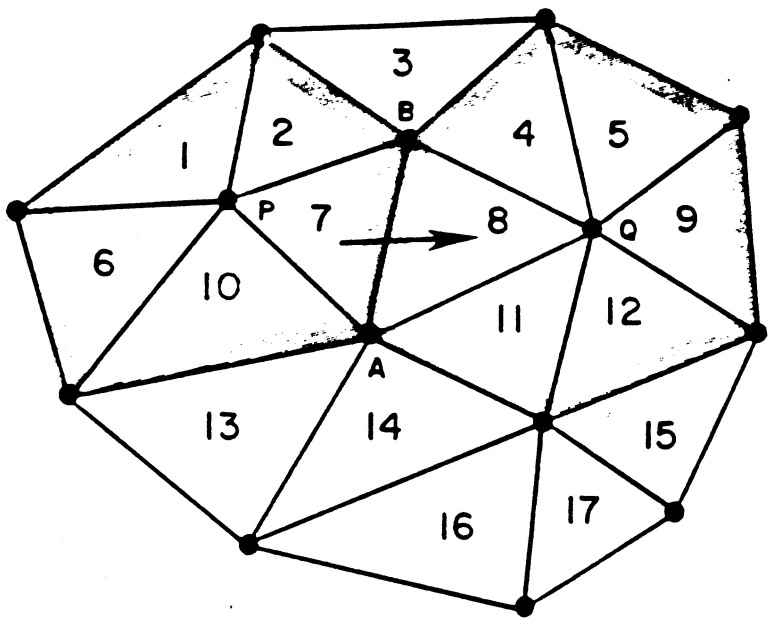


Figure 4

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.

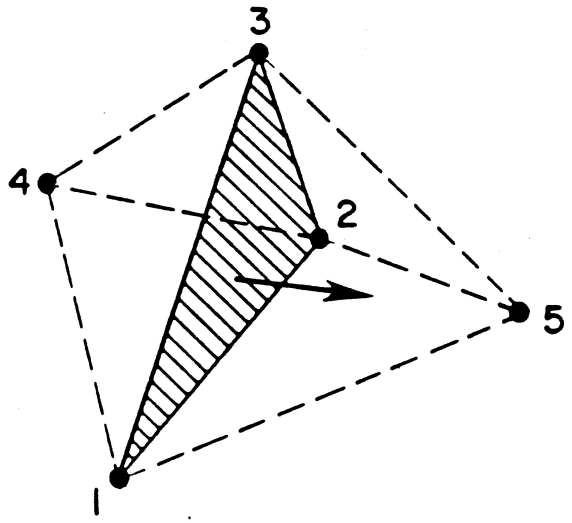


Figure 5

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.

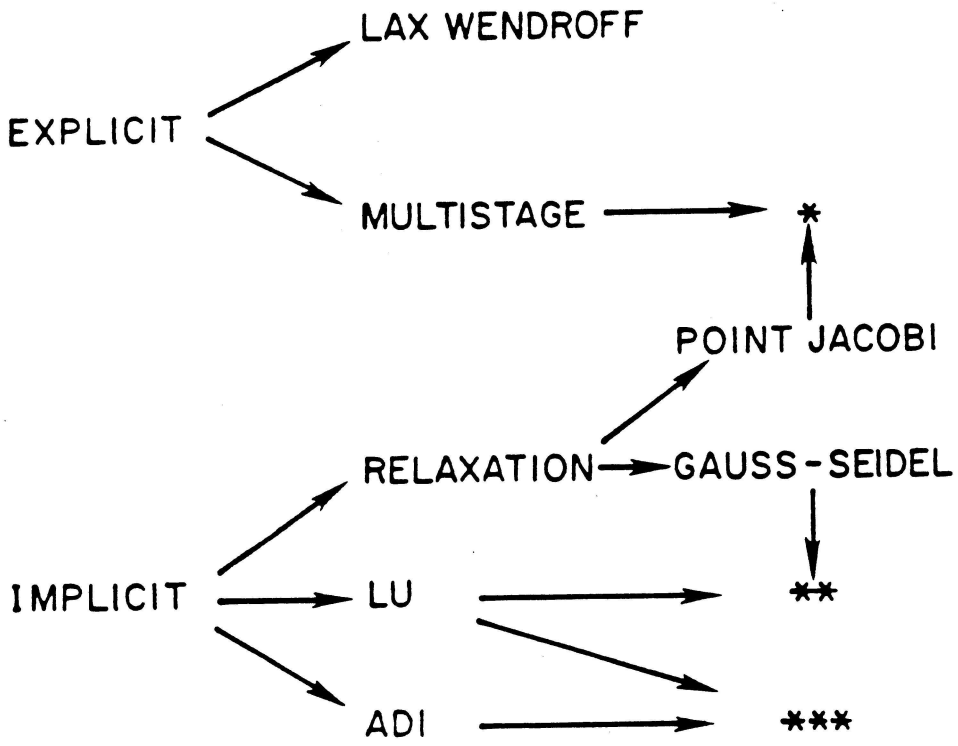
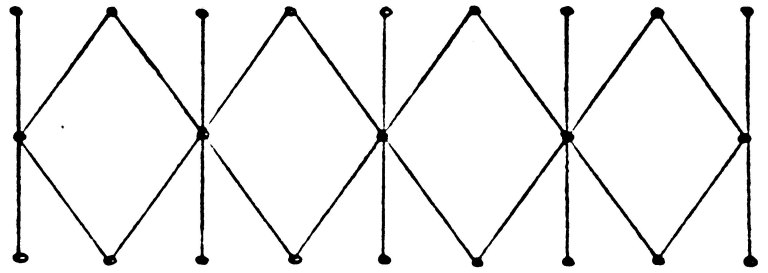


Figure 6

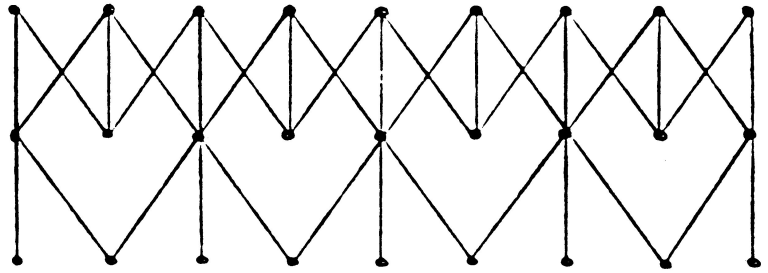
Time Stepping Schemes

* facilitates vector and parallel processing



(a)

Multigrid scheme

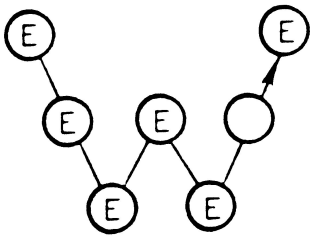


(b)

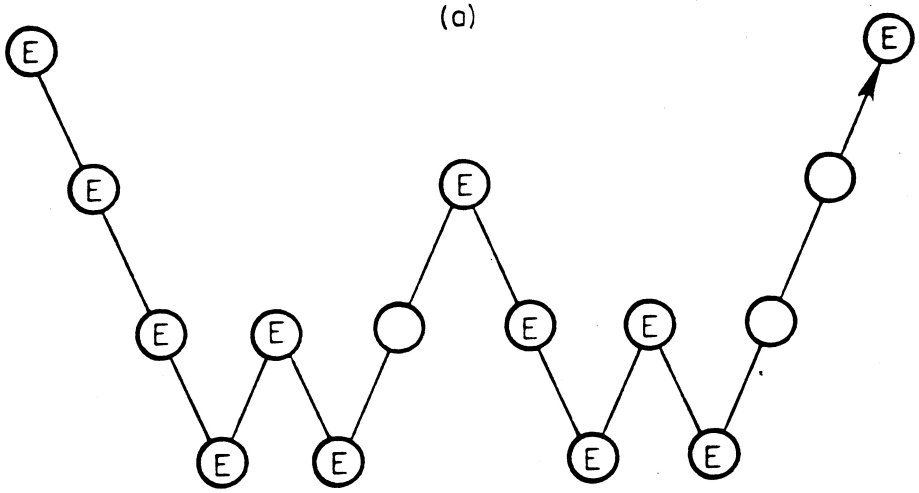
Uniform scheme with nonlinear filter

Figure 7

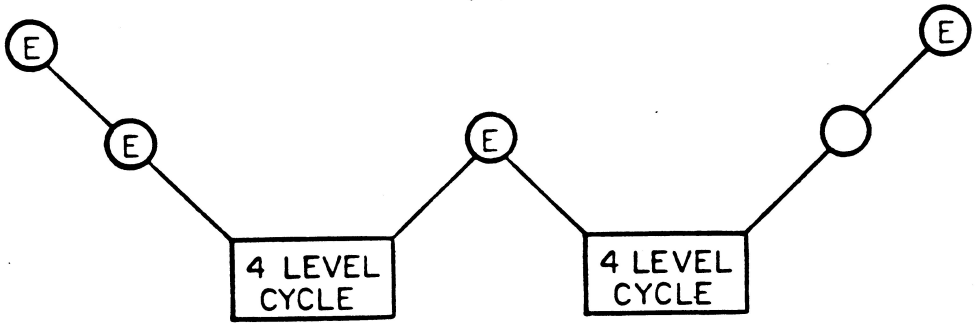
Data Flow of Multigrid and Uniform Schemes



3 LEVELS
(a)



4 LEVELS
(b)



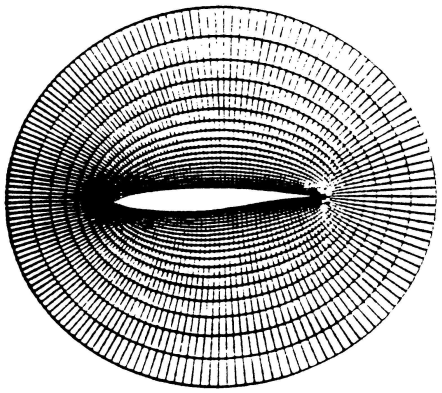
5 LEVELS
(c)

Figure 8

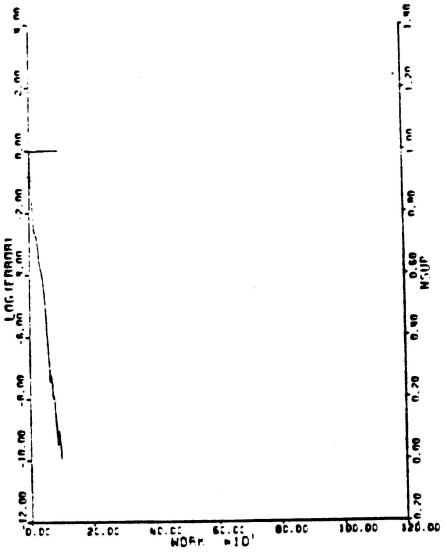
W cycle

E Calculate one time step

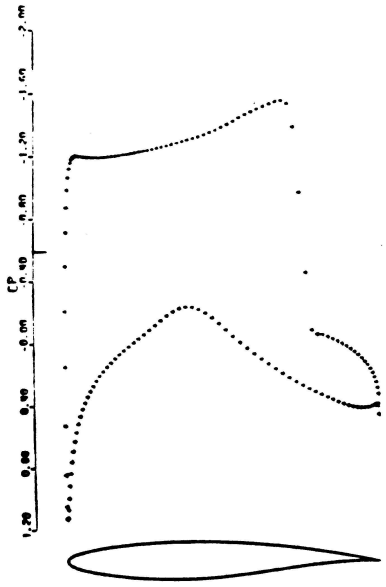
Transfer data without updating the solution.



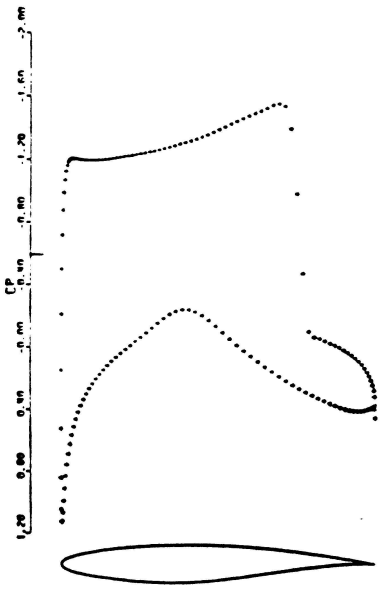
(a) Inner part of the grid



(c) Convergence history
Initial residual .124
Final residual .219 10^{-10}
Average reduction per cycle .797

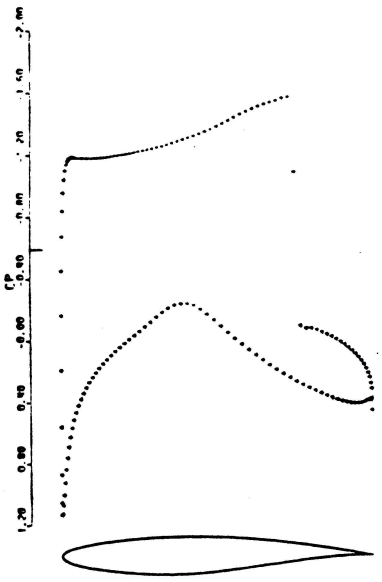


(b) Solution after 100 cycles
CL 1.1256 CD .0470

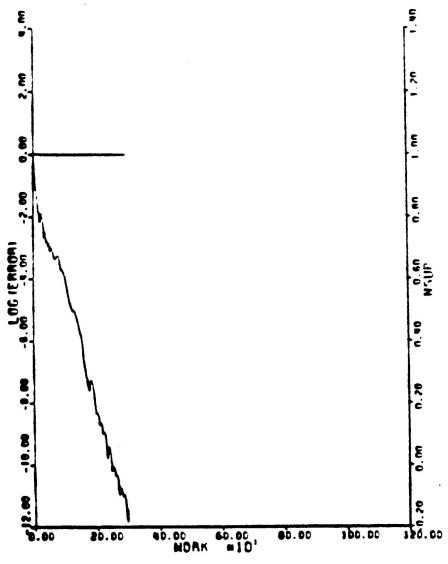


(d) Solution after 10 cycles
CL 1.1258 CD 0470

Figure 9
Euler solution for RAE 2822 airfoil
160x32 grid Mach .750 α 3.0°
Adaptive dissipation

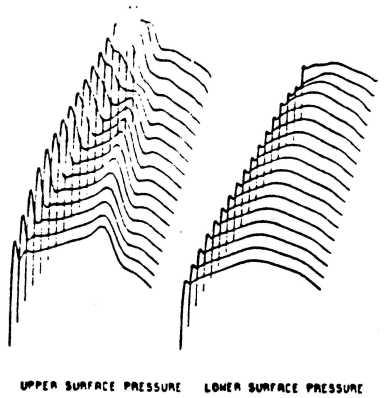


(a) Solution after 300 cycles
CL 1.1155 CD .0481

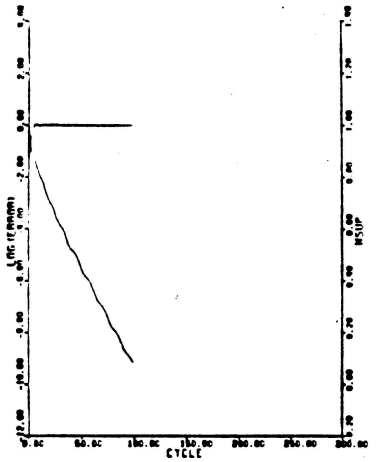


(b) Convergence history
Initial residual .114
Final residual .403 10^{-12}
Average reduction per cycle .516

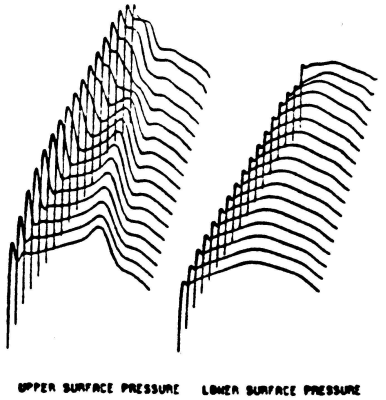
Figure 10
Euler solution for RAE 2822 airfoil
160x32 grid Mach .750 α3.0°
Flux difference split TVD scheme



(a) Solution after 100 cycles
CL .3179 CD .0164

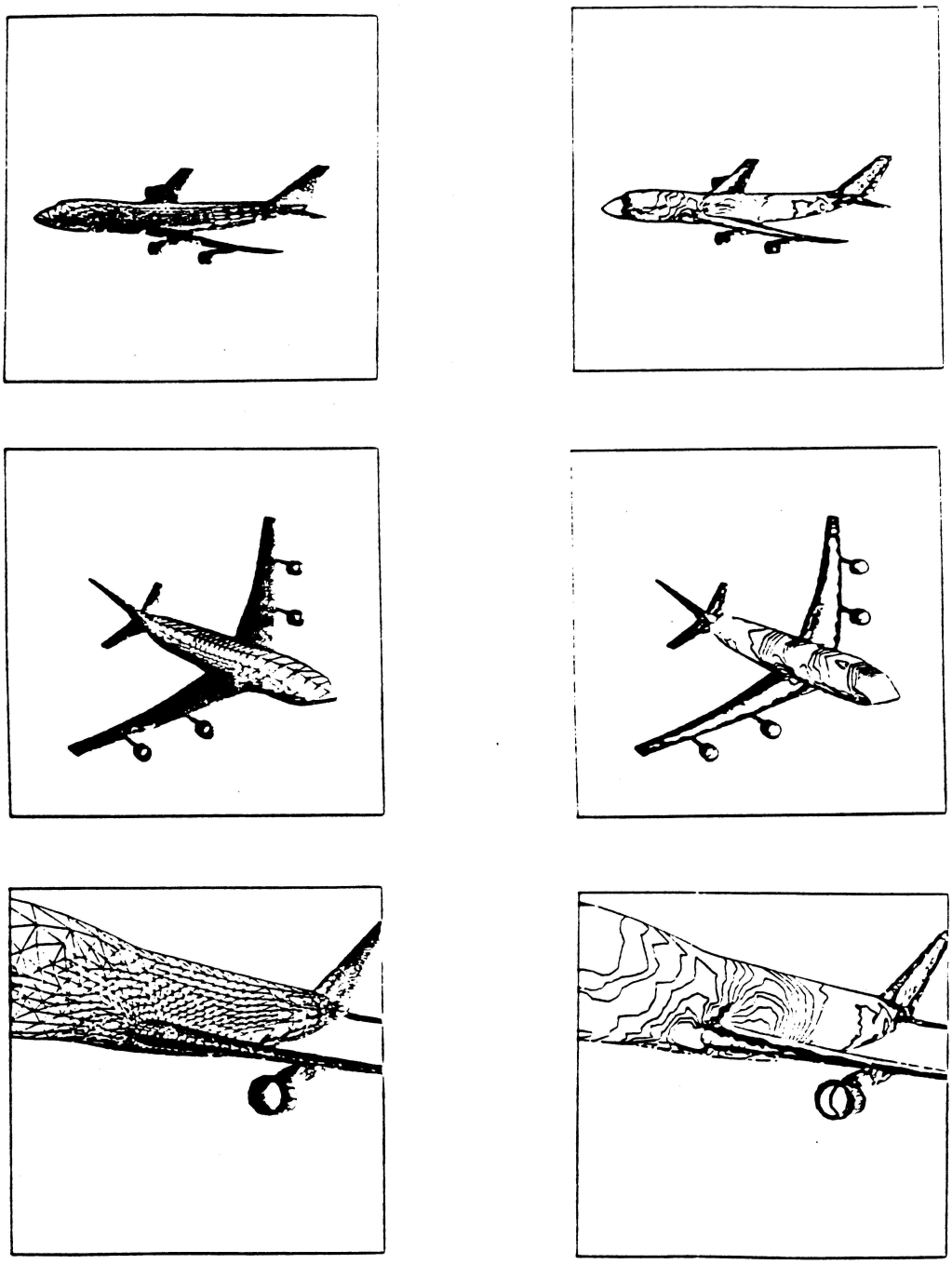


(b) Convergence history
Initial residual .114
Final residual .103 10^{-8}
Average reduction per cycle .808



(c) Solution after 15 cycles
CL .3181 CD .0164

Figure 11
Euler solution for ONERA M6 wing
144x24x24 grid Mach .84 α 3.06°



(a) Views of the mesh

(b) Surface pressure contour

Figure 12
Euler solution for Boeing 747-200
Mach .84 α 2.73°