MULTIGRID ALGORITHMS FOR COMPRESSIBLE FLOW CALCULATIONS

Antony Jameson

October 1985

1 Introduction

During the last two decades computational methods have transformed the science of aerodynamics. Following the introduction of panel methods for subsonic flow in the sixties [1, 2], and major advances in the simulation of transonic flow by the potential flow approximation in the seventies [3–6], the eighties have seen rapid developments in methods for solving the Euler and Navier Stokes equations [7–11]. Multigrid techniques have penetrated this rapidly burgeoning field, with some notable successes [12–21], but they have not yet advanced to the point of gaining general acceptance, nor is their application to non-elliptic problems securely anchored on a firm theoretical foundation. Multigrid time stepping schemes, which are the subject of this paper, have emerged as a promising way to extend multigrid concepts to the treatment of problems governed by hyperbolic equations. The paper falls into two main parts. The first part (Sections 2 and 3) reviews methods of solving the Euler equations of compressible flow, and discusses the trade-offs which underly the choice of schemes for space and time discretization. The second part (Sections 4 and 5) discusses multigrid time stepping schemes, and presents a new method of analyzing the stability of these schemes.

The major considerations in the design of effective methods for the computation of aerodynamic flows are the capability to treat flows over complex geometrical shapes, proper representation of shock waves and contact discontinuities, treatment of viscous effects, and computational efficiency. In practice the viscosity of air is so low that viscous effects are largely confined to thin boundary layers adjacent to the surface. These can only be resolved by the introduction of tightly bunched meshes. At Reynolds numbers typical of full scale flight, of the order of thirty million, the flow in the boundary layer also becomes turbulent, and is unsteady on small scales. The representation of these effects can become prohibitively expensive, and poses a challenge for the future. This paper is restricted to the treatment of inviscid flow. This is already a sufficiently testing problem, and good inviscid methods are needed as a platform for the development of sound viscous methods. The immediate application of the present work is the calculation of steady inviscid transonic flow, and some results are presented for a swept wing. Transonic flow is important because it is the principal operating regime of both commercial and military aircraft. Similar trade-offs, however, apply to a broader class of problems, and the multigrid time stepping technique which I describe should prove useful whenever there is a need to find the steady state solution of a system which is governed by a hyperbolic equation.

The underlying idea is to integrate the time dependent Euler equations until they reach a steady state. In this work the discretization is performed in two stages. The problem is first reduced to a set of ordinary differential equations by subdividing the domain into polygonal or polyhedral cells, and writing the conservation laws in integral form for each cell. The resulting semi-discrete model can then be integrated in time by a variety of discrete time stepping schemes; either implicit or explicit. 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 amalgmated into larger cells which form a coarser mesh. In each coarse 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. This process can be repeated through a sequence of meshes in each of which the mesh spacing is doubled from the previous mesh. 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.

The potential for acceleration through the use of large time steps on the coarse grids is apparent. Its realization in practice is not so easy, and is contingent on ensuring the stability of the composite process, and preventing too much attrition of the convergence rate from the errors introduced by interpolation from coarser to finer grids. The results presented in Section 6, however, indeed confirm the power of the method. In fact, three dimensional solutions of the Euler equations can be obtained in 10-20 multigrid cycles.

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

$$\begin{aligned} E &= \frac{p}{(\gamma-1)\rho} + \frac{1}{2}(u^2+v^2+w^2) \\ H &= E + \frac{p}{\rho} \end{aligned}$$

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

$$\frac{\partial}{\partial t}\iiint\limits_{\Omega} w d\Omega + \iint\limits_{\partial\Omega} \mathbf{F}\cdot\mathbf{dS} = 0 \tag{2.1}$$

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

$$\begin{aligned} w &= \rho \\ \mathbf{F} &= (\rho u, \rho v, \rho w) \end{aligned}$$

For conservation of momentum in the x direction

$$\begin{aligned} w &= \rho u \\ \mathbf{F} &= (\rho u^2 + P, \rho uv, \rho uw) \end{aligned}$$

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

$$\begin{aligned} w &= \rho E \\ \mathbf{F} &= (\rho H u, \rho H v, \rho H w) \end{aligned}$$

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 2.1 for a two-dimensional case.

Figures 2.1(a) and 2.1(b) show cell centered schemes on rectilinear and triangular meshes [7,18]. In either case equation (2.1) 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

$$Q = \sum_k \mathbf{F_{0k}}\cdot\mathbf{S_{0k}} \tag{2.3}$$

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

$$\mathbf{F_{0k}} = \frac{1}{2}(\mathbf{F_0} + \mathbf{F_k}) \tag{2.4}$$

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

$$Q_{0k} = \frac{1}{2}(\mathbf{q_0} + \mathbf{q_k}) \cdot \mathbf{S_{0k}}$$

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

Figures 2.1(c) and 2.1(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

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

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

$$Q = \sum_l \mathbf{F_l} \cdot \mathbf{S_l} \tag{2.6}$$

where $\mathbf{S_l}$ is the directed area of the l^{th} face, and $\mathbf{F_l}$ 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 contribute to its flux balance. The flux balance at each vertex can be evaluated directly by summing either the flux balances of its constituent cells or the contributions of its faces. Alternatively the flux balance at every vertex can be evaluated indirectly by a loop over the faces. In this case the flux $\mathbf{F} \cdot \mathbf{S}$ across a given face is accumulated into the flux balance of those control volumes which contain that face as an external face. On a tetrahedral mesh each face is shared by exactly two control volumes (centered at the outer vertices ofthe two tetrahedra which have that face as a common base, as illustrated in Figure (2.2)), and this is a very efficient procedure [11].

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,

$$\mathbf{F_{12}} = \frac{1}{2}(\mathbf{F_1} + \mathbf{F_2})$$

in Figure 2.1(c) or 2.1(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 (2.3)). Storage of the solution at the vertices has a similar advantage 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

$$\mathbf{F} = \frac{1}{3}(\mathbf{F_1} + \mathbf{F_2} + \mathbf{F_3})$$

is a natural choice, which is consistent with the assumption that $\mathbf{F}$ varies linearly over the face. It can be shown that the resulting scheme is essentially equivalent to the use of a Galerkin method with piecewise linear basis functions [11]. If a rectilinear mesh is used in a three dimensional problem, the difficulty arises that the four corners of a face are not necessarily coplanar. The use of a simple average of the four corner values is a natural choice on a smooth mesh, but in order to preserve accuracy on an irregular mesh it may be necessary to use more complex integration formulas based on local mappings.

The approximations (2.2) or (2.5) need to be augmented by artificial dissipative terms 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

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \tag{2.7}$$

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 (2.7) 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

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

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

$$\frac{d}{dt} v_j = \sum_{-Q}^{Q-1} c_q(j) \left(v_{j-q} - v_{j-q-1} \right) \tag{2.8}$$

these conditions are

$$c_{-1}(j-1) > c_{-2}(j-2) \cdots > c_{-Q}(j-Q) > 0 \tag{2.9a}$$

and

$$-c_0(j) > -c_1(j+1) \cdots > -c_{Q-1}(j+Q-1) > 0 \tag{2.9b}$$

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

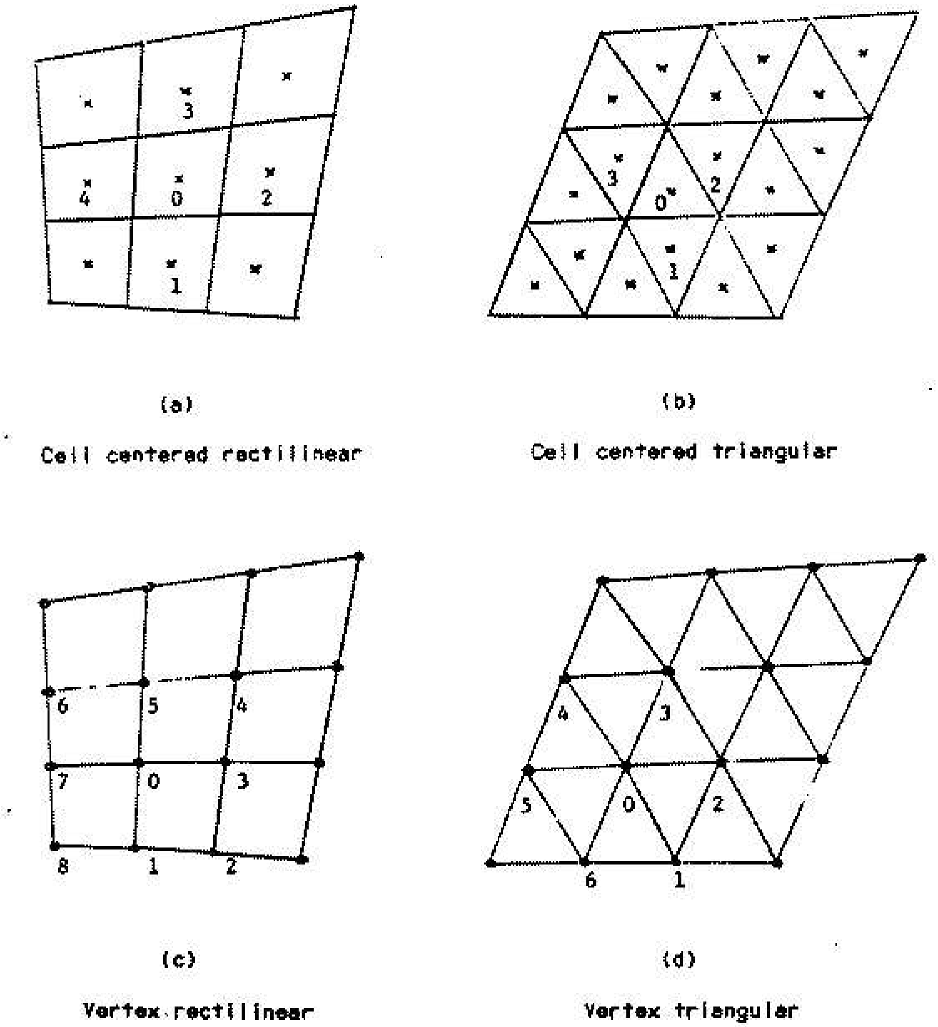
$$\frac{d}{dt} v_j = c_{j+1/2}(v_{j+1} - v_j) - c_{j-1/2}(v_j - v_{j-1})$$

is TVD if $c_{j+1/2} > 0$, $c_{j-l/2} > 0$. A conservative semi-discrete approximation to equation (2.7) 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

$$\Delta x \frac{d}{dt} v_j + h_{j+1/2} - h_{j-1/2} = 0 \tag{2.10}$$

where $h_{j+l/2}$ is the estimate of the flux between cells j and $j+1$. Conditions (2.9) are satisfied by the upwind scheme

$$h_{j+1/2} = \begin{cases} f(v_j) & \text{if } a_{j+1/2} > 0 \\ f(v_j+1) & \text{if } a_{j+1/2} < 0 \end{cases} \tag{2.11}$$



(a)

Cell centered rectilinear

(b)

Cell centered triangular

(c)

Vertex rectilinear

(d)

Vertex triangular

Figure 1

Alternative discretization schemes

Figure 2.1: Alternative Discretization Schemes

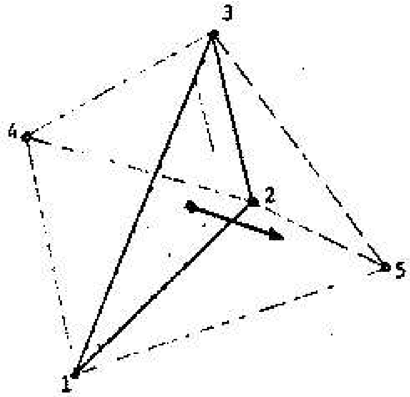


Figure 2

Vertex scheme on a tetrahedral mesh.
Flux through face 1 2 3 influences
balance in control volumes centered at 4 and 5.

Figure 2.2: Vertex Scheme on a Tetrahedral Mesh

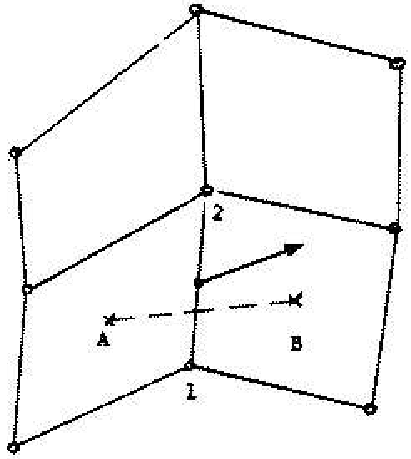


Figure 3

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 than averaging $\underline{F_A}$ and $\underline{F_B}$.

Figure 2.3: Comparison of discretization schemes on a kinked mesh.

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

$$a_{j+1/2} = \begin{cases} \frac{f_{j+1}-f_j}{v_{j+1}-v_j} & \text{if } v_{j+1} \neq v_j \\ \left.\frac{\partial f}{\partial v}\right|_{v=v_j} & \text{if } v_{j+1} = v_j \end{cases} \tag{2.12}$$

More generally, if one sets

$$h_{j+1/2} = \frac{1}{2}(f_{j+1} + f_j) + \alpha_{j+1/2}(v_{j+1} - v_j) \tag{2.13}$$

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

$$\alpha_{j+1/2} \geq \frac{1}{2}\left|a_{j+1/2}\right| \tag{2.14}$$

since one can write

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

and

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

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

$$\mathbf{f}(\mathbf{w_{j+1}}) - \mathbf{f}(\mathbf{w_j}) = A_{j+1/2}(\mathbf{w_{j+1}} - \mathbf{w_j}) \tag{2.15}$$

Roe gives a method of constructing 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 $\mathbf{w_{j+1}} - \mathbf{wj}$ as a sum of the eigenvectors $\mathbf{r_l}$, of $A_{j+1/2}$,

$$\mathbf{w_{j+1}} - \mathbf{w_j} = \sum \alpha_l r_l \tag{2.16}$$

$$\mathbf{f_{j+1}} - \mathbf{f_j} = \sum \lambda_l \alpha_l \mathbf{r_l} \tag{2.17}$$

and the desired dissipative term can be constructed as

$$\sum \mu_l \alpha_l \mathbf{r_l} \tag{2.18}$$

where

$$\mu_l > \frac{1}{2}\left|\lambda_l\right| \tag{2.19}$$

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 VanLeer [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

$$\frac{\partial v}{\partial t}+\frac{\partial}{\partial x}f(v)+\frac{\partial}{\partial y}g(v)=0 \tag{2.20}$$

The mesh may be either rectilinear or triangular, as sketched in Figure 2.1. 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

$$\frac{dv_0}{dt}=\sum_k c_k(v_k-v_0) \tag{2.21}$$

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

$$c_k>0, \quad k=1,2\cdots \tag{2.22}$$

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 equatlon (2.20) can be reduced to the form (2.21) 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 rectilinear mesh, as in Figure 2.1(c). Then equation (2.20) is replaced by

$$S\frac{dv_0}{dt}+\frac{1}{2}\sum_k\left\{\left(f_k+f_{k-1}\right)\left(y_k-y_{k-1}\right)-\left(g_k+g_{k-1}\right)\left(x_k-x_{k-1}\right)\right\}=0 \tag{2.23}$$

where k ranges from 1 to 8. This can be rearranged as

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

where

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

and this is equivalent to

$$S\frac{dv_0}{dt}+\sum_k\left\{(f(v_k)-f(v_0))\Delta y_k-(g(v_k)-g(v_0))\Delta x_k\right\}=0 \tag{2.24}$$

Define the coefficient a_{k0} as

$$a_{k0}=\begin{cases}\frac{(f_k-f_0)\Delta y_k-(g_k-g_0)\Delta x_k}{v_k-v_0}, & v_k\neq v_0\\ \left.\left(\frac{\partial f}{\partial v}\Delta y_k-\frac{\partial g}{\partial v}\Delta x_k\right)\right|_{v=v_0}, & v_k=v_0\end{cases} \tag{2.25}$$

Then equation (2.24) reduces to

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

To produce a scheme satisfying the sign condition (2.22), 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

$$\alpha_{k0}>|a_{k0}| \tag{2.26}$$

The definition (2.25) and condition (2.26) correspond to the definition (2.12) and condition (2.14) 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

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

Then $\mathbf{w_k} - \mathbf{w_0}$ is expanded as a sum of the elgenvectors 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

$$d_{i+1/2,j,k} - d_{i-1/2,j,k} + d_{i,j+1/2,k} - d_{i,j-1/2,k} + d_{i,j,k+1/2} - d_{i,j,k-1/2} \tag{2.27}$$

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

$$d_{i+1/2,j,k} = R(\epsilon^{(2)} - \epsilon^{(4)}\delta_x^2)(\rho_{i+1,j,k} - \rho_{i,j,k}) \tag{2.28}$$

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 by 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 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]).

3 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

$$\frac{d\mathbf{w}}{dt} + \mathbf{R}(\mathbf{w}) = 0 \tag{3.1}$$

where $\mathbf{w}$ is the vector of the flow variables at the mesh points, and $\mathbf{R}(\mathbf{w})$ 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 unsteady problem is used only as a vehicle for reaching the steady state, alternative iterative schemes might be contemplated. Two possibilities in particular are the least squares method, which has been successfully employed to solve a variety of nonlinear problems byGlowinski and his co-workers [6], and the Newton iteration, which has recently been used to solve the two dimensional Euler equations by Giles [37]. The strategy of the present work, however, is to rely upon the simplest possible method, and to attempt to obtain a fast rate of convergence by the use of multiple grids.

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 $\mathbf{R}(\mathbf{w})$. Figure 3.1 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 number of popular explicit schemes are based on the Lax Wendroff formulation, in which the change $\delta\mathbf{w}$ during a time step is calculated from the first two terms of the Taylor series

$$\delta\mathbf{w} = \Delta t \frac{\partial \mathbf{w}}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{w}}{\partial t^2}$$

In a two dimensional case for which $\mathbf{R}(\mathbf{w})$ approximates $\frac{\partial}{\partial x}\mathbf{f}(\mathbf{w}) + \frac{\partial}{\partial y}\mathbf{g}(\mathbf{w})$, the the second derivative is then estimated by substituting

$$\begin{aligned}
\frac{\partial^2 \mathbf{w}}{\partial t^2} &= -\frac{\partial}{\partial t}\mathbf{R}(\mathbf{w}) \\
&= -\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial t} \\
&= \left(\frac{\partial}{\partial x}A + \frac{\partial}{\partial y}B\right)\mathbf{R}(\mathbf{w})
\end{aligned}$$

where A and B are the Jacobian matrices

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{w}}, \quad B = \frac{\partial \mathbf{g}}{\partial \mathbf{w}}$$

In a variation which has been successfully used to drive multigrid calculations by Ni [16], and Hall [20], the flow variables are stored at the cell vertices. The correction at a vertex is then calculated from the average of the residuals in the four neighbouring cells, augmented by differences in the x and y directions of the residuals multiplied by the Jacobian matrices. Accordingly

$$\delta\mathbf{w} = -\left\{\mu_x\mu_y - \frac{\Delta t}{2}\left(\frac{1}{\Delta x}\mu_y\delta_x A + \frac{1}{\Delta y}\mu_x\delta_y B\right)\right\}\Delta t\mathbf{Q}(\mathbf{w}) \tag{3.2}$$

where μ and δ denote averaging and difference operators, and $\mathbf{Q}(\mathbf{w})$ is the flux balance in one cell, calculated in the manner described in Section 2. Ni views this is a rule for distributing a correction calculated at the center of each cell unequally to its four corners. As it stands, the distribution rule is consistent with a steady state solution $\mathbf{Q}(\mathbf{w}) = 0$, but it does not damp a high frequency mode with alternate signs at odd and even points. In shock capturing applications it has proved necessary to augment the correction with a dissipative term $\Delta t\mathbf{D}(\mathbf{w})$. The steady state solution then depends on the time step Δt because the second term in the expansion is multiplied by Δt^2. Consequently the time stepping procedure cannot be regarded as an iterative scheme independent ofthe equations to be solved.

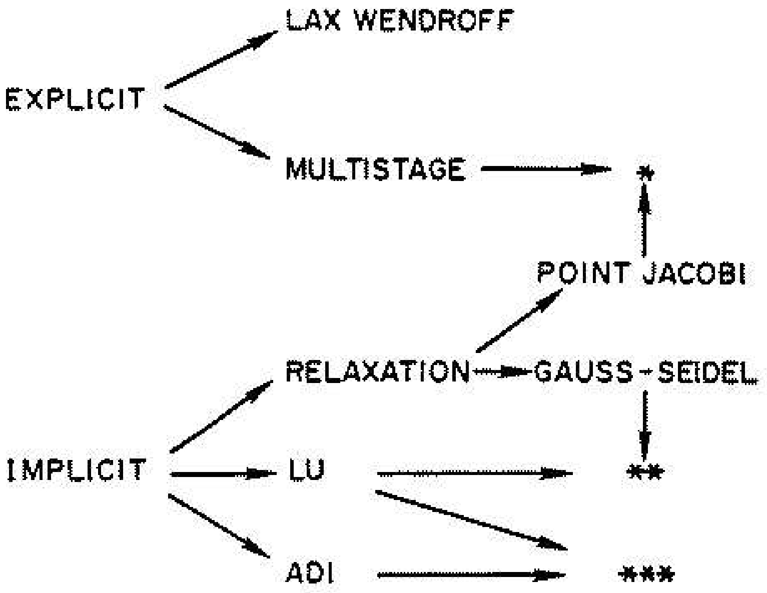


Figure 4

Time stepping schemes

* facilitates vector and parallel processing

Figure 3.1: Time stepping schemes.

The same is true of a number of two step Lax Wendroff schemes in which $\Delta t \frac{\partial \mathbf{w}}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{w}}{\partial t^2}$ is replaced by an estimate of $\frac{\partial \mathbf{w}}{\partial t}$ at the time $t + \frac{\Delta t}{2}$. In the widely used MacCormack scheme, the predictor and corrector steps are

$$(1) \quad \mathbf{w}^* = \mathbf{w^n} - \Delta t \left(D_x^+ \mathbf{f^n} + D_y^+ \mathbf{g^n} \right)$$

and

$$(2) \quad \mathbf{w^{n+1}} = \mathbf{w^n} - \frac{\Delta t}{2} \left(D_x^+ \mathbf{f^n} + D_y^+ \mathbf{g^n} \right) - \frac{\Delta t}{2} \left(D_x^- \mathbf{f}^* + D_y^- \mathbf{g}^* \right)$$

where D_x^+ , D_x^- , D_y^+ and D_y^- are forward and backward difference operators approximating $\partial/\partial x$ and $\partial/\partial y$. Here the use of different approximations for $\partial f/\partial x + \partial g/\partial y$ in the two stages leads to a dependence of the steady state solution on Δt. If one regards equation (3.1) as a set of ordinary differential equations in which $\mathbf{R}(\mathbf{w})$ has a fixed form, then the steady state solution is unambiguously $\mathbf{R}(\mathbf{w}) = \mathbf{0}$. 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 [7, 8, 11].

Let $\mathbf{w^n}$ be the result after n steps. The general form of an m stage scheme is

$$\begin{aligned}
\mathbf{w^{(0)}} &= \mathbf{w^n} \\
\mathbf{w^{(1)}} &= \mathbf{w^{(0)}} - \alpha_1 \Delta t \mathbf{R^{(0)}} \\
\dots & \\
\mathbf{w^{(m-1)}} &= \mathbf{w^{(0)}} - \alpha_{m-1} \Delta t \mathbf{R^{(m-2)}} \\
\mathbf{w^{(m)}} &= \mathbf{w^{(0)}} - \Delta t \mathbf{R^{(m-1)}} \\
\mathbf{w^{n+1}} &= \mathbf{w^{(m)}}
\end{aligned} \tag{3.3}$$

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

$$\mathbf{R^{(q)}} = \sum_{r=0}^{q} \beta_{qr} \mathbf{R}(\mathbf{w^{(r)}}) \tag{3.4a}$$

where

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

In the simplest case

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

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, however, to separate the residual $\mathbf{R}(\mathbf{w})$ into its convective and dissipative parts $\mathbf{Q}(\mathbf{w})$ and $\mathbf{D}(\mathbf{w})$. Then the residual in the $(q+1)st$ stage is evaluated as

$$\mathbf{R^{(q)}} = \sum_{r=0}^{q} \beta_{qr} \mathbf{Q}(\mathbf{w^{(r)}}) - \gamma_{qr} \mathbf{D}(\mathbf{w^{(r)}}) \tag{3.4b}$$

where

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

Blended multistage 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.

Instead of recalculating the residuals at each stage it is possible to estimate the change in the residual by the first term of a Taylor expansion

$$\mathbf{R}(\mathbf{w} + \delta \mathbf{w}) = \mathbf{R}(\mathbf{w}) + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \delta \mathbf{w} + \cdots$$

If $\mathbf{R}(\mathbf{w})$ approximates $\frac{\partial}{\partial x}\mathbf{f}(\mathbf{w})+\frac{\partial}{\partial y}\mathbf{g}(\mathbf{w})$ and the dissipative terms are frozen, this leads to the following reformulation of the multistage scheme:

$$\begin{aligned}
\delta\mathbf{w^{(1)}} &= -\alpha_1\Delta t\mathbf{R}(\mathbf{w^{(0)}})\\
\delta\mathbf{w^{(2)}} &= -\alpha_2\Delta t\mathbf{R}(\mathbf{w^{(0)}})+(D_xA+D_yB)\,\delta\mathbf{w^{(1)}}\\
\delta\mathbf{w^{(3)}} &= -\alpha_3\Delta t\mathbf{R}(\mathbf{w^{(0)}})+(D_xA+D_yB)\,\delta\mathbf{w^{(2)}}\\
\ldots &
\end{aligned}$$

This formulation in which the current estimate of the correction is repeatedly multiplied by the Jacobians to produce an improved estimate, resembles the Lax Wendroff scheme. It differs from the Lax Wendroff scheme, however, in applying the difference operator D_xA+D_yB to the actual $\delta\mathbf{w_j}$ calculated at the mesh points x_j in the previous stage. Accordingly it has the stability properties of the multistage scheme.

The properties of multistage schemes can be further enhanced by residual averaging [8]. 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

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

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

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

where $\Delta t*$ 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

$$(1-\epsilon_x\delta_x^2)(1-\epsilon_y\delta_y^2)(1-\epsilon_z\delta_z^2)\bar{R}=R \tag{3.5}$$

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 [11, 19]. 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 $\partial\mathbf{w}/\partial t$ at $t+\mu\Delta t$ as a linear combination of $\mathbf{R}(\mathbf{w^n})$ and $\mathbf{R}(\mathbf{w^{n+1}})$ The resulting equation

$$\mathbf{w^{n+1}}=\mathbf{w^n}-\Delta t\left\{(1-\mu)\mathbf{R}(\mathbf{w^n})+\mu\mathbf{R}(\mathbf{w^{n+1}})\right\} \tag{3.6}$$

can be linearized as

$$(I+\mu\Delta t\frac{\partial\mathbf{R}}{\partial\mathbf{w}})\delta\mathbf{w}+\Delta t\mathbf{R}(\mathbf{w^n})=0 \tag{3.7}$$

Equation (3.7) 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

$$(I+\mu\Delta tD_xA)(I+\mu\Delta tD_yB)\delta\mathbf{w}+\Delta t\mathbf{R}(\mathbf{w})=0 \tag{3.8}$$

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:

$$\begin{aligned}
(I+\mu\Delta tD_xA)\delta\mathbf{w}^* &= -\Delta t\mathbf{R}(\mathbf{w})\\
(I+\mu\Delta tD_yB)\delta\mathbf{w} &= \delta\mathbf{w}^*
\end{aligned}$$

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 [9], 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 (3.3) by the product of lower and upper block triangular factors L and U,

$$LU\delta\mathbf{w} + \Delta t\mathbf{R}(\mathbf{w}) = 0 \tag{3.9}$$

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 $A = \partial\mathbf{f}/\partial\mathbf{w}$ be split as

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

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

$$L = I + \mu\Delta t D_x^- A^+, \quad U = I + \mu\Delta t D_x^+ A^- \tag{3.10}$$

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 (3.6) or the linearized equation (3.7). A Jacobi method of solving (3.6) can be formulated by regarding it as an equation

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

to be solved for $\mathbf{w}$. Here $\mathbf{w^{(0)}}$ is a fixed value obtained as the result of the previous time step. Now using bracketed superscripts to denote the iterations, we have

$$\begin{aligned}\mathbf{w^{(0)}} &= \mathbf{w^n}\\ \mathbf{w^{(1)}} &= \mathbf{w^{(0)}} + \Delta t\mathbf{R}(\mathbf{w^{(0)}})\end{aligned}$$

and for $k > 1$

$$\mathbf{w^{(k+1)}} = \mathbf{w^{(k)}} + \sigma_{k+1}\left\{\left(\mathbf{w^{(k)}} - \mathbf{w^{(0)}} + \mu\Delta t\mathbf{R}(\mathbf{w^{(k)}}) + (1-\mu)\Delta t\mathbf{R}(\mathbf{w^{(0)}})\right)\right\}$$

where the parameters σ_{k+l} can be chosen to optimize convergence. Finally, if we stop after m iterations,

$$\mathbf{w^{n+1}} = \mathbf{w^{(m)}}$$

We can express $\mathbf{w^{(k+1)}}$

$$\mathbf{w^{(k+1)}} = \mathbf{w^{(0)}} + (1+\sigma_{k+1})(\mathbf{w^{(k)}} - \mathbf{w^{(0)}}) + \sigma_{k+1}\left\{\left(\mu\Delta t\mathbf{R}(\mathbf{w^{(k)}}) + (1-\mu)\Delta t\mathbf{R}(\mathbf{w^{(0)}})\right)\right\}$$

Since

$$\mathbf{w^{(1)}} - \mathbf{w^{(0)}} = \sigma_1\Delta t\mathbf{R}(\mathbf{w^{(0)}})$$

it follows that for all k we can express $(\mathbf{w^{(k)}} - \mathbf{w^{(0)}})$ as a linear combination of $\mathbf{R}(\mathbf{w^{(j)}})$, $j < k$. Thus this scheme is a variant of the multi-stage time stepping scheme described by equations (3.3) and (3.4a). 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 [10,21,45]. Consider the case of a flux split scheme in one dimension, for which

$$\mathbf{R}(\mathbf{w}) = D_x^+ \mathbf{f}^-(\mathbf{w}) + D_x^- \mathbf{f}^+(\mathbf{w})$$

where the flux is split so that the Jacobian matrices

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

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

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

At the jth mesh point this is

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

where

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

Set $\delta \mathbf{w_j^{(0)}} = 0$. A two sweep symmetric Gauss-Seidel scheme 1s then

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

Subtracting (1) from (2) we find that

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

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

$$\begin{aligned} L &= I - \alpha A^- + \mu t D_x^- A^+ \\ U &= I + \alpha A^+ + \mu t D_x^+ A^- \\ D &= I + \alpha (A^+ - A^-) \end{aligned}$$

It follows that the scheme can be written as

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

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 (3.1). 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 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).

Schemes of all three classes have been successfully used in conjunction with multigrid techniques [17–19, 21, 46–48]. 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. The efficiency of this approach has been confirmed in the recent work of Hemker and Spekreijse [21]. 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.

Viewed in the broader context of Runge-Kutta methods for solving ordinary differential equations, the coefficients of a multi-stage scheme can be tailored to optimize the stability region without any requirement of diagonal dominance. As has been noted by Hall, multigrid time stepping methods also expand the domain of dependence of the discrete scheme in a way that corresponds to signal propagation of the physical system. This allows a large effective time step to be attained by a multigrid cycle without the need to introduce an implicit time stepping scheme. The results presented in Section 6 confirm that rapid convergence can indeed be obtained by explicit multi-stage methods in conjunction with a multigrid scheme.

4 Multigrid Time Stepping Schemes

The discrete equations (3.1) describe the local evolution of the system in the neighborhood of each mesh point. The underlying idea of a multigrid time stepping scheme is to transfer some of the task of tracking the evolution of the system to a sequence of successively coarser meshes. 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 large time steps as the process passes to the coarser grids, without violating the stability bound.

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 l 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

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

where $\mathbf{w_{k-1}}$ 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

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

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

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

The result $\mathbf{w_k^{(m)}}$ 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

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

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

$$T_{k,k-1}\mathbf{w_{k-1}} = \frac{\sum V_{k-1}\mathbf{w_{k-1}}}{V_k}$$

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

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

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 $\mathbf{w_k^{(0)}}$ 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:

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

and

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

Then in the three dimensional case

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

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

In this formulation the residuals on each mesh should be re-evaluated after the time step to provide a proper estimate of the current value $\mathbf{R_k}(\mathbf{w_k^+})$ for transfer to the next mesh $k+1$ in the sequence. Just as the multistage time stepping scheme can be modified to eliminate the recalculation of the residuals by substituting a one term Taylor expansion for $\mathbf{R}(\mathbf{w}+\delta\mathbf{w})$, so can the multigrid scheme be modified by a similar substitution to allow the unmodified residuals to be passed to the coarser mesh. This requires the collection operator $Q_{k,k-1}$ to be constructed so that $Q_{k,k-1}\mathbf{R_{k-1}}(\mathbf{w_{k-1}})$ approximates a weighted average of the residuals $\mathbf{R_{k-1}}(\mathbf{w_{k-1}}+\delta\mathbf{w_{k-1}})$. If $\mathbf{R}(\mathbf{w})$ approximates $\partial/\partial x\ \mathbf{f}(\mathbf{w}) + \partial/\partial y\ \mathbf{g}(\mathbf{w})$, and the change in the dissipative term is ignored, $Q_{k,k-1}$ should then be a nonsymmetric operator approximating a multiple of $I + \Delta t_k(D_x A + D_y B)$, where A and B are the Jacobian matrices. Hall uses a procedure of this type in his formulation of a multigrid scheme with Lax Wendroff time stepping [20].

5 Analysis of Multigrid Time Stepping 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. It is proposed here to circumvent this difficulty 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 data from alternate points. For the sake of simplicity the analysis will be restricted to a one dimensional model. It also proceeds on the assumption that each coarser mesh is produced by eliminating alternate points of the finer mesh, so that there exists a set of points which are common to all the meshes.

Figure 5.1(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 $R(u_j)$ is the residual. Here it will be 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 $0, 2, 4...$ are common to both meshes, while the points $1, 3, 5...$ are eliminated in grid 2. A simple multigrid scheme can be described as follows. On grid 1 u_j is updated by a correction

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

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

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

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,

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

where

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

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

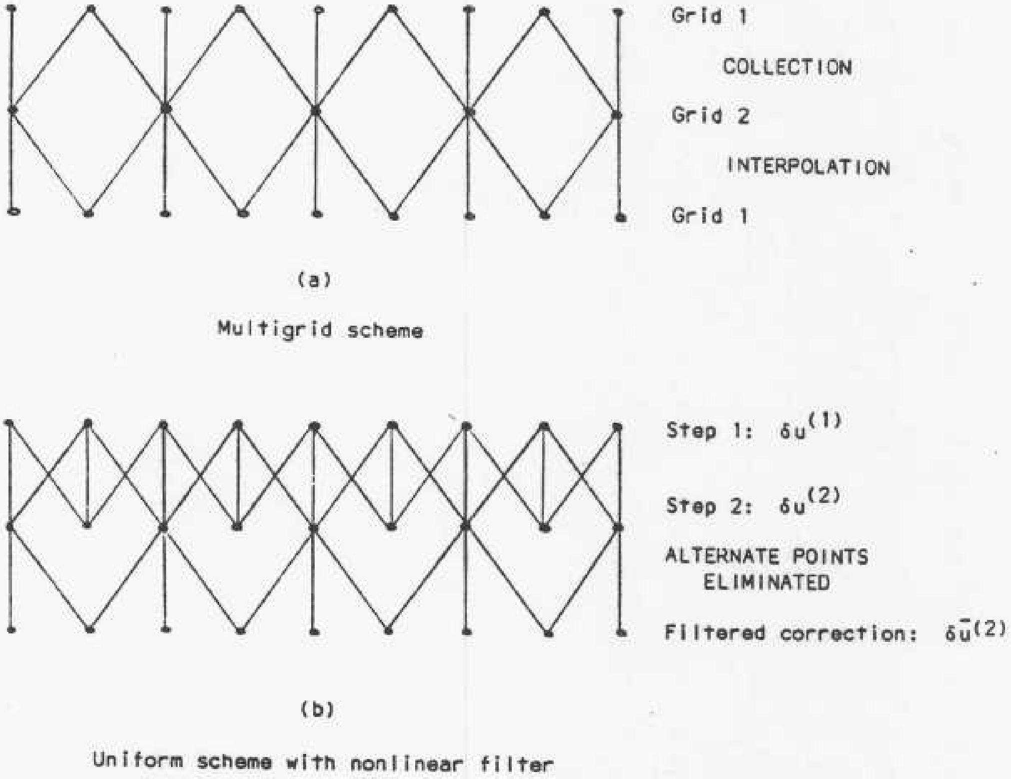
$$\begin{aligned} \delta u_j &= \delta u_j^{(1)} + \delta u_j^{(2)}, \ j \text{ even} \\ \delta u_j &= \delta u_j^{(1)} + \frac{1}{2}(\delta u_{j-1}^{(2)} + \delta u_{j+1}^{(2)}), \ j \text{ odd} \end{aligned}$$

This process can be broken down into two stages as illustrated in Figure 5.1(b). First the corrections $\delta u_j^{(2)}$ are calculated for all points of grid 1 by formulas 5.3,5.4,5.5 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

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

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

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



(a)

Multigrid scheme

(b)

Uniform scheme with nonlinear filter

Figure 5

Data flow of multigrid and uniform schemes

Figure 5.1: Data flow of multigrid and uniform schemes

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

$$e_j = \frac{1}{4}(\delta u^{(2)}_{j-1} - 2\delta u^{(2)}_j + \delta u^{(2)}_{j+1})(1 - \cos\frac{\pi}{\Delta x}x_j) \tag{5.6}$$

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 $\delta u^{(2)}_j$ 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 scheme to have a substantial stability margin at the mesh frequencies of all the meshes above the coarsest mesh in the sequence.

The following paragraphs address the question of the stability of the multilevel uniform scheme. The analysis is carried out for an initial value problem on an infinite interval governed by an equation of the form

$$\frac{\partial \mathbf{v}}{\partial t} + A\mathbf{v} = 0 \tag{5.7}$$

where A is a linear differential operator in one space dimension. The operator A may contain a forcing term, so that $\mathbf{v}$ is not zero when the system reaches a steady state. Let the vector $\mathbf{u}$ with elements $\mathbf{u_j}$ represent the discrete solution. The residual is

$$\mathbf{R} = P\mathbf{u} \tag{5.8}$$

where P is a difference operator approximating $\Delta t A$. In the case of a p^{th} order accurate scheme, if P is applied to the values $\mathbf{v_j} = \mathbf{v}(\mathbf{x_j})$ of the exact solution, then

$$P\mathbf{v} = \Delta t(A\mathbf{v} + O(\Delta x^p))$$

Using supercripts to denote the time steps,

$$\mathbf{u^{n+1}} = \mathbf{u^n} + \delta\mathbf{u}$$

where the correction $\delta\mathbf{u}$ depends on the residual through the action of a time stepping operator F, corresponding to equation 5.2. For example, if we use the multi-stage scheme

$$\begin{aligned}
\mathbf{u^{(0)}} &= \mathbf{u^n} \\
\mathbf{u^{(1)}} &= \mathbf{u^{(0)}} - \alpha_1 P\mathbf{u^{(0)}} \\
\mathbf{u^{(2)}} &= \mathbf{u^{(0)}} - \alpha_2 P\mathbf{u^{(1)}} \\
\mathbf{u^{(3)}} &= \mathbf{u^{(0)}} - \alpha_3 P\mathbf{u^{(2)}} \\
\mathbf{u^{n+1}} &= \mathbf{u^{(3)}}
\end{aligned}$$

we find that

$$\mathbf{u^{(3)}} = \mathbf{u^{(0)}} - \alpha_3(I - \alpha_2 P + \alpha_2\alpha_1 P^2)P\mathbf{u^{(0)}}$$

Consequently

$$F = \alpha_3(I - \alpha_2 P + \alpha_2\alpha_1 P^2)$$

For the Crank Nicolson scheme

$$\mathbf{u^{n+1}} = \mathbf{u^n} - \frac{1}{2}(P\mathbf{u^{n+1}} + P\mathbf{u^n})$$

we obtain

$$F = (I + \frac{1}{2}P)^{-1}$$

If we set

$$\hat{\mathbf{u}}(\xi) = \Delta x \sum_{-\infty}^{\infty} \mathbf{u_j} e^{-i\xi x_j/\Delta x}$$

then the Fourier transform of the residual (5.8) is $\hat{P}\hat{u}$ where $\hat{P}(\xi)$ is the Fourier symbol of the difference operator. Suppose, for example, that

$$A = a\frac{\partial}{\partial x}$$

and that we use a central difference scheme with added dissipative terms. Then

$$(P\mathbf{u})_j = \frac{\lambda}{2}(\mathbf{u_{j+1}} - \mathbf{u_{j-1}}) - \lambda\mu_2(\mathbf{u_{j+1}} - 2\mathbf{u_j} + \mathbf{u_{j-1}}) + \lambda\mu_4(\mathbf{u_{j+2}} - 4\mathbf{u_{j+1}} + 6\mathbf{u_j} - 4\mathbf{u_{j-1}} + \mathbf{u_{j-2}}) \tag{5.9}$$

where λ is the Courant number,

$$\lambda = a\frac{\Delta t}{\Delta x}$$

and μ_2 and μ_4 are dissipative coefficients. Also

$$\hat{P}(\xi) = \lambda i \sin\xi + 2\lambda\mu_2(1-\cos\xi) + 4\lambda\mu_4(1-\cos\xi)^2$$

Similarly if $\hat{F}(\xi)$ is the Fourier symbol of the time stepping operator, then

$$\delta\hat{\mathbf{u}}(\xi) = -\hat{F}(\xi)\hat{P}(\xi)\hat{\mathbf{u}}^{\mathbf{n}}(\xi)$$

and

$$\hat{\mathbf{u}}^{\mathbf{n+1}}(\xi) = g(\xi)\hat{\mathbf{u}}^{\mathbf{n}}(\xi) \tag{5.10}$$

where $g(\xi)$ is the amplification factor

$$g(\xi) = I - \hat{F}(\xi)\hat{P}(\xi) \tag{5.11}$$

Suppose that we have a nested set of grids with successively doubled mesh intervals. It is now convenient to revert to denoting the grids by subscripts $1, 2, 3\cdots$ (Since the individual elements of the solution vector do not appear in the analysis this leads to no confusion). Consider a multigrid time stepping scheme in which time steps are taken on successive grids sequentially down to the coarsest grid, and the cycle is then repeated. In order to produce the same final steady state as a scheme using only the fine grid, the evolution on every grid except grid 1 should driven by the residuals calculated on the next finer grid. Let $\mathbf{R_1^+}$ be the residual on grid 1 after the change $\delta\mathbf{u_1}$ and let $\mathbf{R_2}$ be the residual calculated on grid 2. Also let Q_{21} be the operator transferring residuals from grid 1 to grid 2, so that $Q_{21}\mathbf{R_1}$ is a weighted sum of fine grid residuals corresponding to the coarse grid residual $\mathbf{R_2}$. Then on grid 2 replace $\mathbf{R_2}$ by

$$\bar{\mathbf{R}}_\mathbf{2} = \mathbf{R_2} + \mathbf{S_2}$$

where

$$S_2 = Q_{21}\mathbf{R_1^+} - \mathbf{R_2}$$

and on grid 3 replace $\mathbf{R_3}$ by

$$\bar{\mathbf{R}}_\mathbf{3} = \mathbf{R_3} + \mathbf{S_3}$$

where

$$\begin{aligned}\mathbf{S_3} &= Q_{32}\mathbf{R_2^+} - \mathbf{R_3}\\ &= Q_{32}(Q_{21}\mathbf{R_1^+} + \mathbf{R_2^+} - \mathbf{R_2}) - \mathbf{R_3}\end{aligned}$$

With a single stage time stepping scheme $\delta\mathbf{u_2}$ is determined by substituting the corresponding fine grid residual $Q_{21}\mathbf{R_1^+}$ for $\mathbf{R_2}$. but $\mathbf{R_2}$ needs to be calculated because $\mathbf{R_2^+} - \mathbf{R_2}$ appears in $\mathbf{S_3}$. With a multi-stage time stepping scheme $\mathbf{R_2}$ would be recalculated several times while $\mathbf{S_2}$ would be frozen at its initial value on grid 2. If we examine the action of an m stage scheme on one of the coarser grids, we have

$$\begin{aligned}\mathbf{u_k^{(0)}} &= \mathbf{u_{k-1}^+}\\ \mathbf{u_k^{(1)}} &= \mathbf{u_k^{(0)}} - \alpha_1(Q_{k,k-1}\mathbf{R_{k-1}^+})\\ \mathbf{u_k^{(2)}} &= \mathbf{u_k^{(0)}} - \alpha_2(\mathbf{R_k^{(1)}} + Q_{k,k-1}\mathbf{R_{k-1}^+} - \mathbf{R_k^{(0)}})\\ &\cdots\\ \mathbf{u_k^{(m)}} &= \mathbf{u_k^{(0)}} - (\mathbf{R_k^{(m-1)}} + Q_{k,k-1}\mathbf{R_{k-1}^+} - \mathbf{R_k^{(0)}})\\ \mathbf{u_k^+} &= \mathbf{u_k^{(m)}}\end{aligned}$$

Here in the second stage

$$\begin{aligned}\mathbf{R}_{\mathbf{k}}^{(\mathbf{1})}-\mathbf{R}_{\mathbf{k}}^{(\mathbf{0})} &= P_k(\mathbf{u}_{\mathbf{k}}^{(\mathbf{1})}-\mathbf{u}_{\mathbf{k}}^{(\mathbf{0})}) \\ &= -\alpha_1 P_k Q_{k,k-1}\mathbf{R}_{\mathbf{k-1}}^{+}\end{aligned}$$

whence

$$\mathbf{u}_{\mathbf{k}}^{(\mathbf{2})}-\mathbf{u}_{\mathbf{k}}^{(\mathbf{0})}=-\alpha_2(I-\alpha_1P_k)Q_{k,k-1}\mathbf{R}_{\mathbf{k-1}}^{+}$$

Following through the remaining stages, we find that

$$\delta\mathbf{u}_{\mathbf{k}}=\mathbf{u}_{\mathbf{k}}^{(\mathbf{m})}-\mathbf{u}_{\mathbf{k}}^{(\mathbf{0})}=-F_kQ_{k,k-1}\mathbf{R}_{\mathbf{k-1}}^{+} \tag{5.12}$$

where F_k is the time stepping operator on grid k as it would appear for a single grid.

Now consider the evolution of all quantities in the multigrid process, assuming that it is uniformly applied at every mesh point of grid 1. Suppose that the collection operators Q_{21}, Q_{32} all have the same generic form. On the fine grid denote this by Q, with corresponding Fourier symbol $\hat{Q}(\xi)$. For example, if

$$(Q\mathbf{R})_j=\frac{1}{2}\mathbf{R}_{\mathbf{j-1}}+\mathbf{R}_{\mathbf{j}}+\frac{1}{2}\mathbf{R}_{\mathbf{j+1}} \tag{5.13}$$

then

$$\hat{Q}(\xi)=1+\cos\xi$$

On grid 1 denote the Fourier symbols of the residual and time stepping operators by

$$p_1=\hat{P}(\xi),\quad f_1=\hat{F}(\xi) \tag{5.14a}$$

and the symbol of the first collection operator by

$$q_{21}=\hat{Q}(\xi) \tag{5.14b}$$

For a system of equations these symbols will be matrices. On the subsequent levels the corresponding symbols are

$$p_k=\hat{P}(2^{k-1}\xi),\quad f_k=\hat{F}(2^{k-1}\xi) \tag{5.14c}$$

and

$$q_{k,k-1}=\hat{Q}(2^{k-1}\xi) \tag{5.14d}$$

Now on the first grid

$$\delta\hat{\mathbf{u}}_{\mathbf{1}}=-f_1\mathbf{r}_{\mathbf{1}}$$

where $\mathbf{r}_{\mathbf{1}}$ is the Fourier transform of the residual

$$\mathbf{r}_{\mathbf{1}}=p_1\hat{\mathbf{u}}_{\mathbf{1}}$$

On subsequent grids it follows from equation (5.12) that

$$\delta\hat{\mathbf{u}}_{\mathbf{k}}=-f_k\mathbf{r}_{\mathbf{k}}$$

where

$$\mathbf{r}_{\mathbf{k}}=q_{k,k-1}\mathbf{r}_{\mathbf{k-1}}^{+}$$

Since the system is linear

$$\mathbf{r}_{\mathbf{k-1}}^{+}=\mathbf{r}_{\mathbf{k-1}}+p_{k-1}\delta\hat{\mathbf{u}}_{\mathbf{k-1}}$$

(but in general $\mathbf{r}_{\mathbf{k-1}}^{+}$ is not equal to $p_{k-1}\mathbf{u}_{\mathbf{k-1}}^{+}$ when $k>2$). Substituting for $\delta\hat{\mathbf{u}}_{\mathbf{k-1}}$ we find that

$$\mathbf{r}_{\mathbf{k}}=q_{k,k-1}(I-p_{k-1}f_{k-1})\mathbf{r}_{\mathbf{k-1}} \tag{5.15}$$

Finally for an m level scheme

$$\hat{\mathbf{u}}_{\mathbf{m}}^{(+)}=\hat{\mathbf{u}}_{\mathbf{1}}=\sum_{1}^{m}f_k\mathbf{r}_{\mathbf{k}} \tag{5.16}$$

Equations (5.14,5.15,5.16) define the stability of the complete multilevel scheme. The final formula may be evaluated directly as a sum in which each new term is obtained recursively from the previous term, or as a nested product by the loop

$$Z_m = f_m$$

for $k = m - 1$ to 1.

$$Z_k = f_k + z_{k+1} q_{k+1,k}(I - p_k f_k)$$

and

$$\hat{\mathbf{u}}_{\mathbf{m}} = (1 - Z_1 p_1)\hat{\mathbf{u}}_{\mathbf{1}}$$

If the operators F and P commute, then equation (5.15) may be simplified by the substitution

$$I - p_k f_k = I - f_k p_k = g_k$$

where g_k is the amplification factor of the basic time stepping scheme applied on level k. This will be the case for any scheme applied to a scalar equation, and for typical multi-stage schemes applied to a system of equations.

In the special case that

$$Q_{k,k-1} P_{k-1} = P_k$$

for example, if at the j^{th} mesh point

$$\begin{aligned} \mathbf{R}_{\mathbf{j}} &= \frac{\lambda}{2}(\mathbf{u}_{\mathbf{j+1}} - \mathbf{u}_{\mathbf{j-1}}) \\ (Q\mathbf{R})_j &= \mathbf{R}_{\mathbf{j-1}} + \mathbf{R}_{\mathbf{j+1}} \end{aligned}$$

equation (5.16) reduces to

$$\hat{\mathbf{u}}_{\mathbf{m}}^{(+)} = g_m g_{m-1} \cdots g_1 \hat{\mathbf{u}}_{\mathbf{1}}$$

In general it does not. This result can be proved by noting that

$$\mathbf{r}_{\mathbf{2}} = q_{21}\mathbf{r}_{\mathbf{1}}^{(+)} = q_{21} p_1 \hat{\mathbf{u}}_{\mathbf{1}}^{(+)} = p_2 \hat{\mathbf{u}}_{\mathbf{1}}^{(+)} = p_2 \hat{\mathbf{u}}_{\mathbf{2}}$$

and

$$\mathbf{r}_{\mathbf{2}}^{(+)} = p_2 \hat{\mathbf{u}}_{\mathbf{2}} + p_2 \delta \hat{\mathbf{u}}_{\mathbf{2}} = p_2 \hat{\mathbf{u}}_{\mathbf{2}}^{+}$$

Then

$$\mathbf{r}_{\mathbf{3}} = q_{32}\mathbf{r}_{\mathbf{2}}^{(+)} = q_{32} p_2 \hat{\mathbf{u}}_{\mathbf{2}}^{(+)} = p_3 \hat{\mathbf{u}}_{\mathbf{2}}^{(+)} = p_3 \hat{\mathbf{u}}_{\mathbf{3}}$$

and so on. Consequently it follows that

$$\hat{\mathbf{u}}_{\mathbf{k}}^{(+)} = (I - f_k q_{k,k-1} p_{k-1})\hat{\mathbf{u}}_{\mathbf{k-1}}^{(+)} = g_k \hat{\mathbf{u}}_{\mathbf{k-1}}^{(+)}$$

Formulas (5.14,5.15,5.16) can easily be evaluated for any particular choices of residual operator, time stepping operator and collection operator with the aid of a computer program. Figures 5.2 and 5.3 show typical results for the dissipative central difference scheme (5.9), with the collection operator (5.13). Both results are for blended multi-stage time stepping schemes of the class defined by equations (3.3) and (3.4b). Figure 5.2 shows the amplification factor of a three stage scheme in which the dissipative terms are evaluated once. The Courant number is 1.5 and the coefficients are

$$\begin{aligned} &\alpha_1 = 0.6, \quad \alpha_2 = 0.6 \\ &\beta_{qq} = 1, \quad \beta_{qr} = 0, \quad q > r \\ &\gamma_{q0} = 1, \quad \gamma_{qr} = 0, \quad r > 0 \end{aligned} \tag{5.17}$$

As the number of levels is increased the stability curve defined by the amplification factor is compressed to the left, retaining a large margin of stability at all high frequencies. Thus the scheme should be resistant to the injection of interpolation errors. Figure 5.3 shows the amplification factor of a five stage scheme in which the dissipative terms are evaluated twice. In this case. the coefficients are

$$\begin{aligned} \alpha_1 = 1/4, \quad \alpha_2 = 1/6, \quad \alpha_3 = 3/8, \quad \alpha_4 = 1/2 \\ \beta_{qq} = 1, \quad \beta_{qr} = 0, \quad q > r \\ \gamma_{00} = 1, \quad \gamma_{q1} = 1, \quad \gamma_{qr} = 0, \quad r \neq 1 \end{aligned} \tag{5.18}$$

Residual averaging is also included with a coefficient of .75, and the Courant number is 7.5. Although the stability curve exhibits a bump, there is still a substantial margin of safety, and this scheme has proved very effective in practice [39].

The formulas of this section can be modified to allow for alternative multigrid strategies, including more complicated V and W cycles. Nor is it necessary to use the same time stepping and residual operators on every grid. It may pay, for example, to use a simplified lower order scheme on the coarse grids. This method of analysis, in which the multigrid process is regarded as a multilevel uniform process on a single grid, subject to the injection of additional interpolation errors, is also easily extended to two and three dimensional problems.

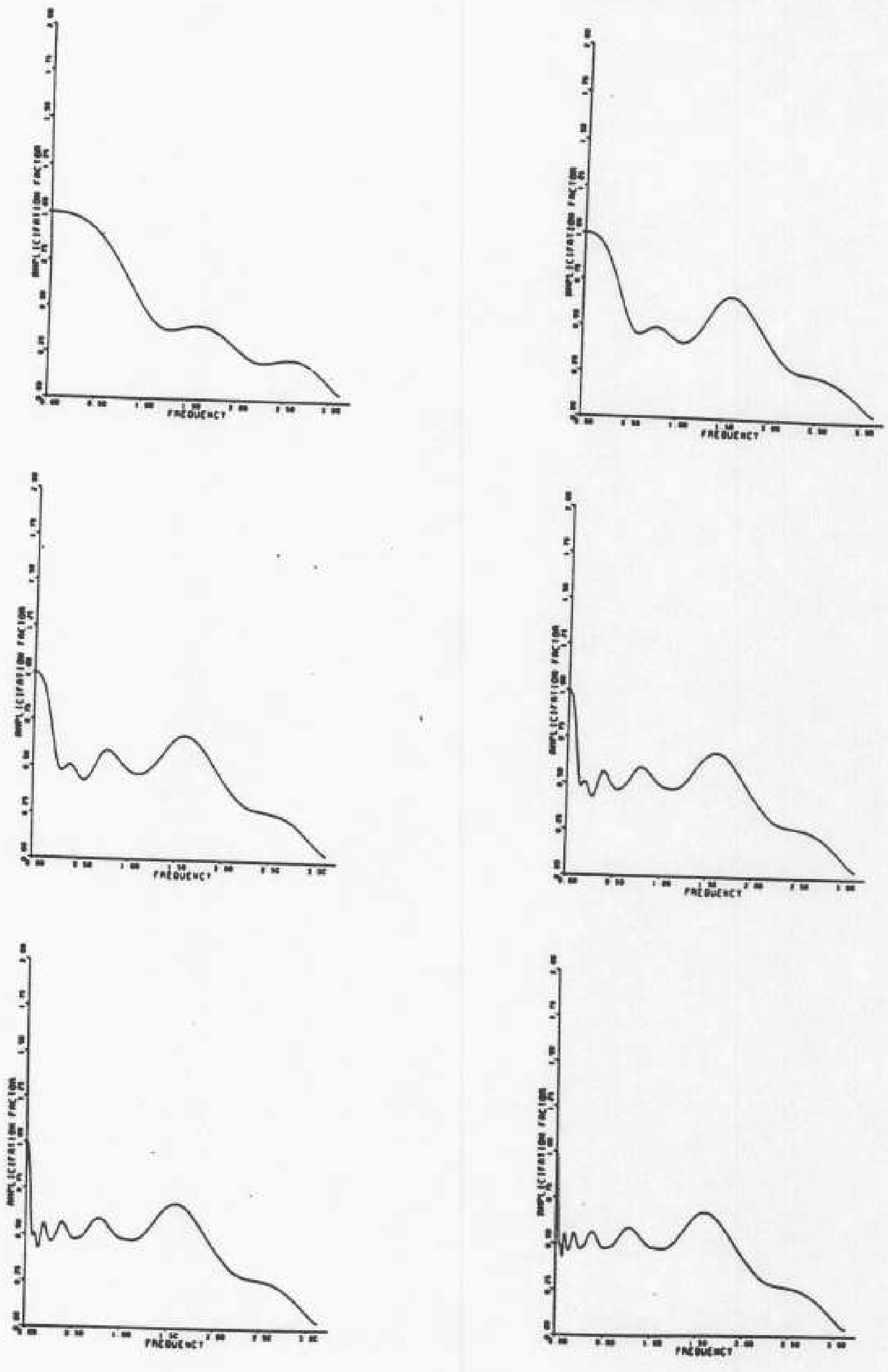


Figure 6

Amplification Diagrams for a 3 Stage Scheme
for 1-6 Grid Levels

Figure 5.2: Amplification Diagrams for a 3 Stage Scheme for 1-6 Grid Levels

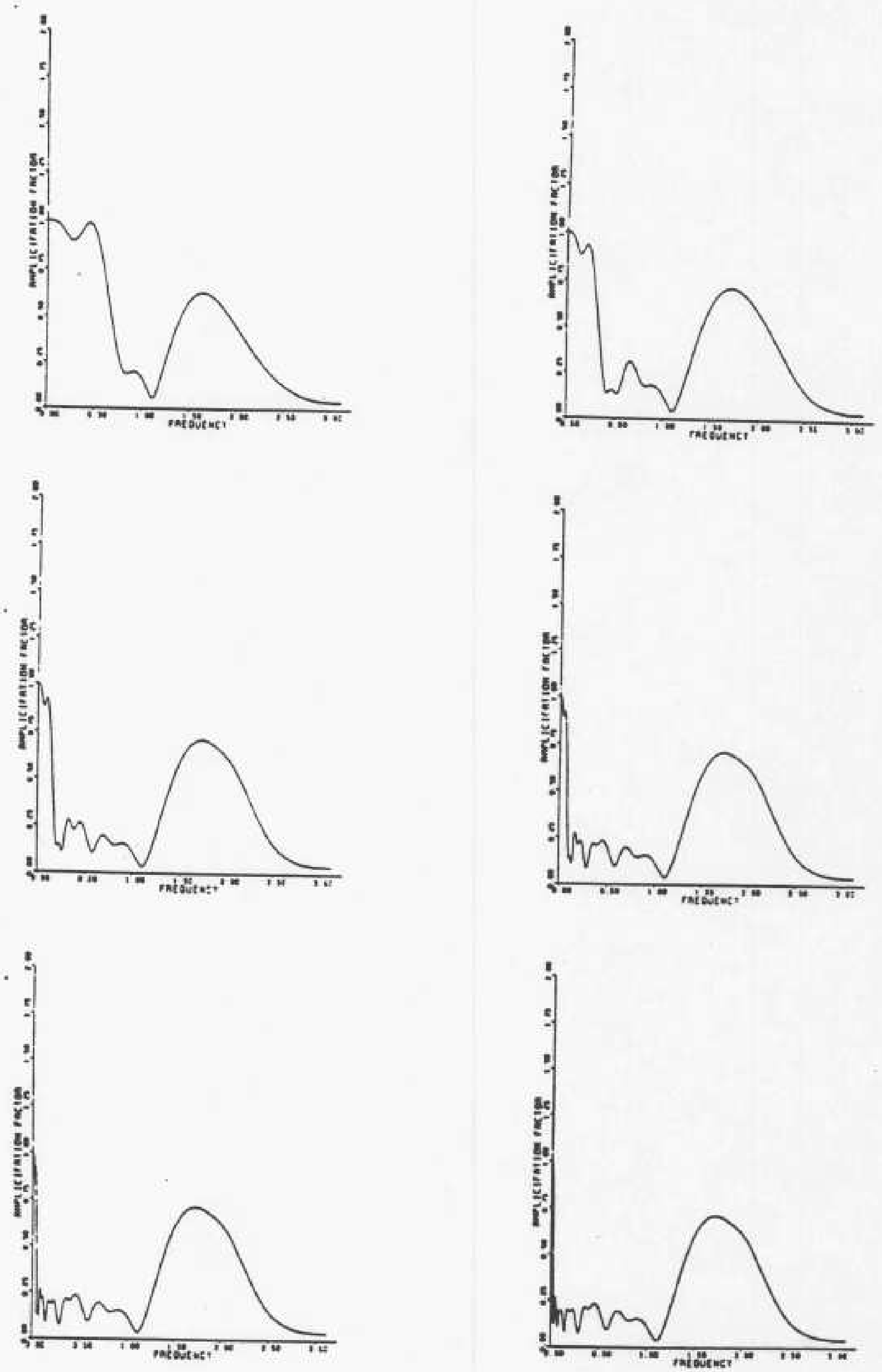


Figure 7

Amplification Diagrams for a 5 Stage Scheme with 2 Evaluations of the Dissipative Terms and Residual Averaging for 1-6 Grid Levels

Figure 5.3: Amplification Diagrams for a 5 Stage Scheme with 2 Evaluations of the Dissipative Terms and Residual Averaging for 1-6 Grid Levels

6 Some Results for an Explicit Multi-stage Scheme

This section presents some results for a simple multigrid method in which an explicit multistage scheme was used for time stepping. The application is the 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 with the coefficients defined by equations (5.18) was used in conjunction with a simple saw tooth multigrid cycle. Implicit residual averaging as defined by equation (3.5) was also used.

The mesh was of C type in streamwise vertical planes, generated by the introduction of sheared parabolic coordinates. This was accomplished by a two stage mapping procedure. The first stage introduces parabolic coordinates by the transformation

$$\begin{aligned}(\bar{X}+i\bar{Y})^2 &= x-x_0(z)+i(y-y_0)/t(z)\\ \bar{Z} &= z\end{aligned}$$

where z is the spanwise coordinate, $t(z)$ is a scaling factor which can be used to control the number of cells covering the wing, and $x_0(z)$ and $y_0(z)$ are the coordinates of a singular line lying just inside the leading edge. The effect of this transformation is to unwrap the wing to a shallow bump $Y = S(X,Z)$. The second stage is a shearing transformation

$$X = X, \quad Y = Y - S(X,Z), \quad Z = Z$$

which maps the wing to the coordinate surface $Y = 0$. The mesh is then constructed by the reverse sequence of mappings from a rectangular grid in the X, Y, Z coordinate system. Meshes of this type contain badly distorted cells in the neighborhood of the singular line where it passes into the flowfield beyond the wing tip. These cells, which have a very high aspect ratio and a triangular cross section, present a severe test of the robustness of the multigrid scheme.

Figure 6.1 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 [1]. 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 6 shows the rate of convergence over 100 multigrid cycles on the $96x16x16$ mesh, 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 6 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, confirming that convergence well within the discretization error can be obtained in $5-10$ cycles. In assessing these results it should be noted that the computational effort of one step of the 5 stage scheme is substantially greater than that of a Lax Wendroff scheme but appreciably less than that required by an alternating direction or LU decomposition scheme. Measured by a work unit consisting of the computational effort of one time step on the fine grid, the work required for one multigrid cycle with five levels is

$$1+\frac{1}{8}+\frac{1}{64}+\frac{1}{512}+\frac{1}{4096}$$

plus the work required for additional residual calculations, which is of the order of 25 percent. Using a single processor of a Cray XMP computer, the time required for a multigrid cycle on a $96x16x16$ mesh is about 1.3 seconds, and a complete solution on such a mesh can be obtained in about 15 seconds. This is fast enough that iteractive analysis of alternative wing designs at the prelimi nary design stage is now within the realm of possibility.

[1] 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.

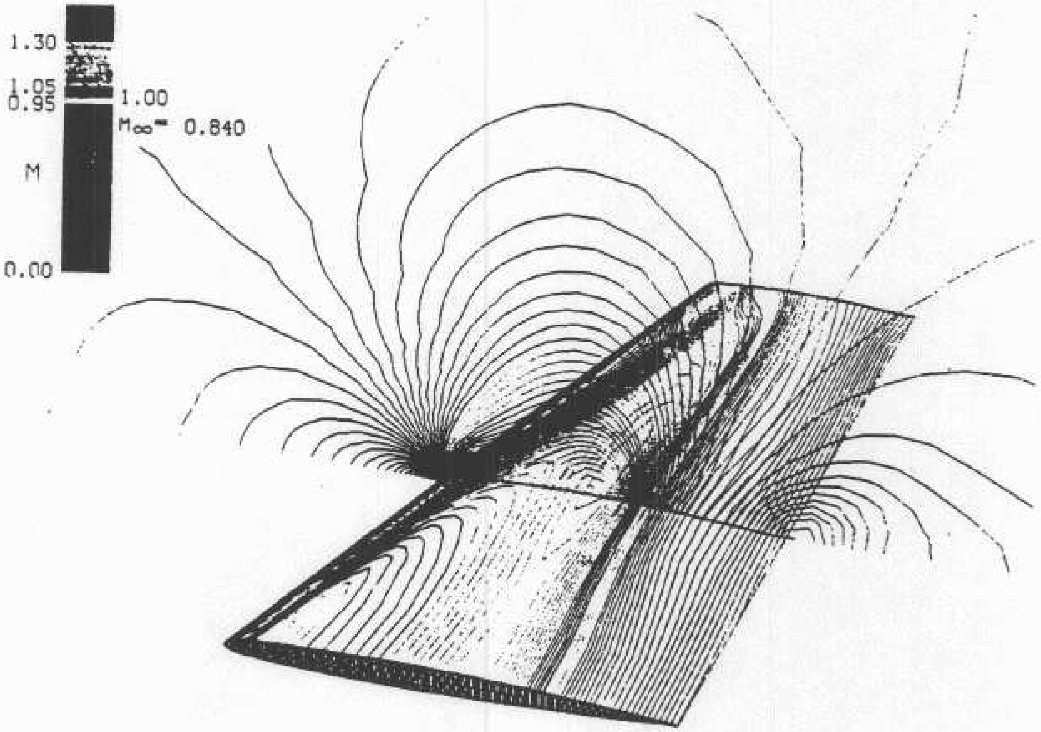


Figure 8

Constant pressure contours of flow over the ONERA M6 wing

Figure 6.1: Constant pressure contours of flow over the ONERA M6 wing

Cycle	Average $d\rho/dt$	CL	CD
1	.916 10^{-1}		
10	.158 10^{-2}	0.3110	0.0205
20	.243 10^{-3}	0.3118	0.0203
30	.245 10^{-4}	0.3118	0.0203
40	.353 10^{-5}	0.3118	0.0203
50	.528 10^{-6}	0.3118	0.0203
60	.772 10^{-7}	0.3118	0.0203
70	.124 10^{-8}	0.3118	0.0203
80	.241 10^{-9}	0.3118	0.0203
90	.363 10^{-9}	0.3118	0.0203
100	.528 10^{-10}	0.3118	0.0203

Table 1: Calculation of the flow past the ONERA M6 wing at Mach 0.840, and 3.06^0 angle of attack on a $96x16x16$ mesh. Average reduction of $d\rho/dt$ per multigrid cycle: 0.807

	CL	CD
Result after 10 cycles on $32x8x8$ mesh	.2956	.0373
Result after 10 cycles on $64x16x16$ mesh	.3167	.0263
Result after 5 cycles on $128x32x32$ mesh	.3145	.0167
Final Converged Result on a $128x32x32$ mesh	.3144	.0164

Table 2: Result for the ONERA M6 Wing with a sequence of 3 meshes

7 Conclusions

Multigrid techniques for the Euler equations are by now solidly established, and a variety of rapidly convergent methods have been demonstrated. The concept of a multigrid time stepping scheme provides an alternative framework for the analysis of these methods. In contrast to the more classical view of the multigrid process based upon assumptions of ellipticity, this concept emphasizes the role of the coarse grids in increasing the speed at which disturbances can be propagated through the domain. It leads rather naturally to the method of analysis proposed in Section 5, which may prove useful for screening alternative multigrid strategies, and identifying those which are most promising.

While the successes which have been achieved to date are enough to indicate the potential of multigrid methods, much work remains to be done. Several particularly important topics of investigation may be singled out. First, the extreme geometrical complexity of the configurations which need to be treated in many engineering applications may well dictate the use of patched and unstructured meshes. The use of an unstructured tetrahedral mesh appears, for example, to be one of the more promising ways to calculate the flow past a complete aircraft [11]. If multigrid methods are to be more widely used, I believe, therefore, that it will be necessary to develop effective methods for unstructured meshes. Second, accurate simulations of real flows must include the effects of viscosity and turbulence, and will accordingly require the treatment of the Reynolds averaged Navier Stokes equations. The need to use meshes with very high aspect ratio cells in the boundary layer region accentuates the difficulties in obtaining rapid convergence. While some acceleration has been demonstrated with multigrid techniques, the speed of convergence still falls far short of the rates achieved in Euler calculations. A third direction of improvement which needs to be pursued is the integration of multigrid solution strategies with procedures for automatic grid refinement. Results which have already been obtained in two dimensional calculations clearly show the potential advantages of such an approach, which could be the key to better resolution of both shock waves and boundary layers [49, 50].

The realization of these improvements will bring us closer to the ultimate goal of accurate and economical prediction of flows over complete con figurations. Computational methods may then finally fulfill their proper role as a reliable guide for the design of aeroplanes, cars, and any other devices whose performance significantly depends on aerodynamic efficiency.

References

[1] J. L. Hess and A. M. 0. Smith. Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies. *Douglas Aircraft Report, ES 40622*, 1962.

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

[3] E. M. Murman and J.D. Cole. Calculation of Plane Steady Transonic Flows. *AIAA Journal, Vol. 9*, pages 114–121, 1971.

[4] Antony Jameson. Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1. *Comm. Pure. Appl. Math, Vol. 27*, pages 283–309, 1974.

[5] Antony Jameson and D. A. Caughey. A Finite Volume Method for Transonic Potential Flow Calculations. *Proc. AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque*, pages 35–54, 1977.

[6] Bristeau, M. 0., Pironneau, 0., 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*, pages 363–394, 1984. edited by J. St. Doltsinis, North Holland, 1985.

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

[8] Antony Jameson and Timothy J. Baker. Solution of the Euler Equations for Complex Configurations. *Proc. AlAA 6th Computational Fluid Dynamics Conference, Danvers*, pages 293–302, 1983.

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

[10] R.W. MacCormack. Current Status of Numerical Solutions of the Navier-StokesEquations. *AIAA Paper 85-0032, AIAA 23rd Aerospace Sciences Meeting, Reno*, January 1985.

[11] Jameson, A., Baker, T.J., and Weatherill, N.P. Calculation of Inviscid Transonic Flow Over a Complete Aircraft. *AlAA Paper 86-0103, AlAA 24th Aerospace Sciences Meeting, Reno*, January 1986.

[12] R. P. Fedorenko. The Speed of Convergence of One Iterative Process. *USSR Comp. Math. and Math. Phys., Vol. 4*, pages 227–235, 1964.

[13] South J. C. and A. Brandt. Application of a Multi-Level Grid Method to Transonic Flow Calculations. *Proc. of Workshop on Transonic Flow Problems in Turbomachinery, Monterey, 1976*, pages 180–206, 1977. edited by T. C. Adamson and M. F. Platzer, Hemisphere.

[14] Antony Jameson. Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method. *Proc. AIAA 4th Computational Fluid Dynamics Conference, Williamsburg*, pages 180–206, 1979.

[15] D. A. Caughey. Multigrid Calculation of Three-Dimensional Transonic Potential Flows. *AlAA Paper 83-0374, AIAA 21st Aerospace Sciences Meeting, Reno*, January 1983.

[16] Ron Ho. Ni. A Multiple Grid Scheme for Solving the Euler Equations. *AIAA Journal, Vol. 20*, pages 1565–1571, 1982.

[17] A. Jameson. Solution of the Euler Equations by a Multigrid Method. *Applied Math. and Computation, Vol. 13*, pages 327–356, 1983.

[18] Jameson, A., and Schmidt, W. Recent Developments in Numerical Methods for Transonic Flows. *Proc. 3rd International Conference on Finite Elements in Nonlinear Mechanics, FENOMECH 84, Stuttgart*, pages 467–493, 1984. edited by J. St. Doltsinis, North-Holland,1985.

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

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

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

[22] S. K. Godunov. A Difference Method for the Numerical Calculation of Discontinous Solutions of Hydrodynamic Equations. *Mat. Sbornik, 47*, pages 271–306, 1959. 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*, pages 38–69, 1973.

[24] B. Van Leer. Towards the Ultimate Conservative Difference Scheme. II, Monotonicity and Conservation Combined in a Second Order Scheme. *J. Comp. Phys. Vol. 14*, pages 361–370, 1974.

[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*, pages 263–293, 1981.

[26] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comp. Phys., Vol. 43*, pages 357–372, 1981.

[27] S. Osher and F. Solomon. Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws. *Math. Comp., Vol. 38*, pages 339–374, 1982.

[28] A. Harten. High Resolution Schemes for Hyperbolic Conservation Laws. *J. Comp. Phys., Vol. 49*, pages 357–393, 1983.

[29] Stanley Osher and Sukumar Chakravarthy. High Resolution Schemes and the Entropy Condition. *SIAM J. Num. Analysis, Vol. 21*, 1984.

[30] P. K. Sweby. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM J. Num. Analysis, Vol. 21*, pages 995–1011, 1984.

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

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

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

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

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

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

[37] M. Giles, M. Drela, and W.T. Thompkins. Newton Solution of Direct and Inverse Transonic Euler Equations. *AlAA Paper 85-1530, Proc. AlAA 7th Computational Fluid Dynamics Conference, Cincinnati*, pages 394–402, 1985.

[38] I.P.E. Kinmark. 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*, pages 345–349, 1984.

[39] A. Jameson. Transonic Flow Calculations for Aircraft. *Lecture Notes in Mathematics, Vol. 1127, Numerical Methods in Fluid Dynamics, edited by F. Brezzi, Springer Verlag*, pages 156–242, 1985.

[40] A.R. Gourlay and A.R. Mitchell. A Stable Implicit Difference Scheme for Hyperbolic Systems in Two Space Variables. *Numer. Math., Vol. 8*, pages 367–375, 1966.

[41] R.W. Beam and R.F. Warming. An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Form. *J. Comp. Phys., Vol. 23*, pages 87–110, 1976.

[42] A. Jameson and Turkel E. Implicit Schemes and LU Decompositions. *Math. Comp. Vol. 37*, 1981.

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

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

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

[46] A. Jameson and S. Yoon. Multigrid Solution of the Euler Equations Using Implicit Schemes. *AlAA Paper 85-0293, AIAA 23rd Aerospace Sciences Meeting, Reno*, January, 1985.

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

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

[49] M. Berger and A. Jameson. Automatic Adaptive Grid Refinement for the Euler Equations. *AlAA Journal, Vol. 23*, pages 561–568, 1985.

[50] J.F. Dannenhoffer and J.R. Baron. Robust Grid Adaption for Complex Transonic Flows. *AlAA Paper 86-0495, AIAA 24th Aerospace Sciences Meeting, Reno*, January 1986.