AIAA 81-0991R

Some Prospects for the Future of Computational Fluid Dynamics

Harvard Lomax
NASA Ames Research Center, Moffett Field, Calif.

Introduction

IT is certainly clear that computer speeds and computer memories have been increasing at a rapid rate over the past two decades. Statistics gathered by Chapman[1] show an increase by a factor of 10 about every seven years for both of these computer properties. A question arises whether improvements in numerical techniques can be expected to show a similar trend. Of course, one must consider how such improvements are to be measured. Three criterion are suggested. One obvious measure is the convergence rate for a given accuracy. A second important but more nebulous measure is reliability, that is, the property of providing correct answers on a variety of problems; it is sometimes referred to as robustness. A third criterion, and one that is especially important in three dimensions, is the ease with which a technique can be adapted to complicated geometries. Improvements are foreseen in all three of these categories and a discussion of some of the techniques that form the basis for this belief is presented in the body of the paper.

A great deal of work has been done during the last decade on the analysis of transonic flows, using the velocity potential equations. One can look forward to some improvements in these codes, mostly in the areas of reliability and adaptability in the three-dimensional cases. However, the discussion in this paper is directed toward the study of two different sets of problems. One involves the analysis of numerical methods directed toward the solution of the equations governing the conservation of mass, momentum, and energy in the flow of an inviscid gas; these equations are referred to as the compressible Euler equations, or, more loosely, simply as the Euler equations, since we are not considering incompressible flows. The other involves the analysis of the compressible Reynolds-averaged, Navier-Stokes equations.

In the 1950's and 1960's, several finite-difference methods were developed for solving the Euler equations in conservation-law form, using the concept of "shock capturing" rather than "shock fitting." The most popular of these were the Lax-Wendroff types,[2] the most widely used of which was MacCormack's explicit method.[3]

As computer resources grew, attempts were made to improve the solution found by these explicit methods by refining the space meshes in the regions of shocks, stagnation points, and boundary surfaces. Also the viscous terms in the Navier-Stokes equations began to be added. Since the convergence rates of explicit methods are highly sensitive to space mesh size, the computer time required to remove the transient terms from the solutions become excessive. This motivated the inclusion of implicit techniques into the codes. The 1970's saw the development and use of factored implicit schemes, such as those due to Briley and McDonald[4] and Beam and Warming,[5] and the development and use of the hybrid MacCormack scheme.[6]

Including some form of implicit procedure in the solution process causes a one to two order of magnitude increase in the convergence rates over the fully explicit methods. Forms of numerical instability still appear, however, and the codes need considerable improvement in convergence rates, reliability, and geometric adaptability. Recently, attempts have been made to improve the first two of these measures by 1) making use of the local eigensystem, and 2) using forms of multigrid tactics developed for elliptic equations. Attempts to improve the third measure are being made through the use of, among other things, zonal methods.

The discussion ends with some remarks on the language requirements for the CFD community.

Some Notation

In the following, use is made of some notation that is convenient but not necessarily conventional. We are concerned with the study of fluid flow by advanced scientific computers, which make use of array processing, and some of our notation reflects this fact. There are two types of vectors that have to be represented in the course of the discussion. One relates to the dependent variables appearing in the partial differential equations. These vectors are of order two to five, depending on the number of dimensions and the kind of approximation. When analyzing the Euler equations the basic notation is

Q= dependent variable vector

F= flux vector

C= Jacobian coefficient matrix

Harvard Lomax received his B.A. (Mechanical Engineering, 1944) and M.A. (Engineering Science, 1947) degrees from Stanford University. He has been employed as a Research Scientist at Ames Research Center since 1944, originally under NACA, and subsequently under NASA. He has been Chief of the Computational Fluid Dynamics Branch since it was formed in 1970. He is a member of the Tau Beta Pi and Phi Beta Kappa honor societies. He received the NASA medal for exceptional scientific achievement in 1973 and the AIAA award in Fluid and Plasma Dynamics in 1977. He has authored or co-authored over 60 technical papers on analytical and numerical aerodynamic research. He became a Fellow of AIAA in 1978.

Presented as Paper 81-0991 at the 5th Computational Fluid Dynamics Conference, Palo Alto, Calif., June 22-23, 1981; submitted June 22, 1981.

$X^{-1}, X =$ left and right eigenvector matrices of C

$x'_m, x_m = m$th row of X^{-1} and column of X

$$\lambda = \text{eigenvalues of } C \tag{1}$$

These may be in conservative or nonconservative form, depending on the discussion in which they are contained. Since there are a limited number of symbols in the English language, these notations may be assigned different meanings, in which case they are locally redefined in the text.

The other type of vector relates to the discrete representation of a group of dependent variables at points through space. The basic notation for this space vector is

$$(Z) \equiv (Z_1, Z_2, \ldots, Z_j, \ldots, Z_{MJ})^T \tag{2}$$

where Z might represent Q, F, or C and j is a space index.

It is also convenient to use a banded-matrix notation.[7] Let $B(\ldots, d_{-1}, d_0, d_1, \ldots)$ designate a banded matrix in which the arguments represent the diagonals. The number of arguments of B is always odd; the middle argument designates the central diagonal, arguments to the left of central designate consecutively lower diagonals, and those to the right, consecutively higher diagonals. For example, $B(1)$ is the identity matrix I, and a five-banded matrix is given by the equation

$$B(d_{-2}, d_{-1}, d_0, d_1, d_2) \equiv \begin{bmatrix} d_0 & d_1 & d_2 & 0 & 0 \\ d_{-1} & d_0 & d_1 & d_2 & \\ d_{-2} & d_{-1} & d_0 & & \\ 0 & d_{-2} & & & \\ 0 & & & & \text{etc.} \end{bmatrix} \tag{3}$$

Two generalizations need to be made if the arguments are not scalar constants. The notation $B(b_{-1}, b_0, b_1)$ represents a tridiagonal matrix in which the diagonals are vectors of scalars. If the diagonals are block matrices, the argument separator is a semicolon, and a scalar argument implies a block-identity matrix. Thus,

$$B(1;-2;1)(C) \equiv \begin{bmatrix} -2C_1 & C_2 & & \\ C_1 & -2C_2 & C_3 & \\ & C_2 & -2C_3 & \\ & & & \ddots \end{bmatrix} \tag{4}$$

Finally, the subscript p on a banded matrix means that the matrix is circulant, which implies that a multiplying space vector is periodic.

Algorithm Development for the Euler Equations

In a later section, in which the computation of high Reynolds number flows is discussed, it is suggested that many practical flowfields that are strongly affected by regions of separated turbulent flow could be computed now, with existing turbulence models, and that the results would be very useful to a qualified aerodynamic engineer. However, this is not being done because of the expense in both computer and calendar time. Lengthy computer times are caused by the very slow convergence rates of existing codes; lengthy calendar time is a result of the fact that the existing codes tend to "blow up" when flowfield parameters are extended out of their checked-out range, and sophisticated studies are then required to find out why. The latter property is referred to as lack of code robustness. For these reasons algorithm development is still a critical area for the computational fluid dynamics field.

In this section, some recent trends for devising new methods that might improve convergence and robustness in numerical solutions of the Euler equations are considered. The discussion is limited entirely to finite-difference or finite-volume techniques. This does not mean that the future of finite element[8] and spectral[9] methods in this area can be neglected. In fact, if the communication barrier can be broken (perhaps through common matrix identifications), the three disciplines may well serve to complement one another.

Use of Local Eigensystems

In the last few years several finite-difference schemes have been developed that make use of the eigenvalues and eigenvectors that belong to the coefficient matrices of the Euler equations. Specifically, they find the eigenvectors associated with each eigenvalue and develop schemes that make use of this information. Conceptually, these methods are related to the method of characteristics, but in application they are quite different, and in detail they differ from one another. There is reason to believe they will add to code robustness and possibly to code efficiency. It appears, therefore, that the details of making them more and more practical will be a fruitful source of material for the process of method development in the future.

A method that proceeded along these lines was presented in 1952 by Courant et al.[10] There were later developments by, among others, Steger,[11] Moretti,[12] Chakravarthy et al.,[13] Steger and Warming,[14] Van Leer,[15] and MacCormack.[16] The reasoning behind the development of these various techniques involves various combinations of physics and numerical analysis and is quite different from method to method. However, the unifying theme is the exploitation of the local eigensystem. The purpose here is to show some of the similarities that exist between the methods and how they relate formally, though not in application, to more "conventional" methods.

For illustrative purposes we choose for a model equation the isothermal form of the Euler equations in one dimension and in conservation-law form

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{5}$$

Let $c^2 = p/\rho$ be the constant isothermal wave speed and set

$$Q = (\rho, \rho u)^T \equiv (q_1, q_2)^T$$

$$F = (\rho u, \rho u^2 + \rho c^2)^T = \left(q_2, \frac{q_2^2}{q_1} + c^2 q_1\right)^T \equiv (f_1, f_2)^T \tag{6}$$

Notice that the flux vector is homogeneous and of degree one, which is true of the Euler equations in general. Because of the homogeneous property $F = CQ$, where C is the Jacobian of F given by

$$C = \frac{\partial f_i}{\partial q_j} = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix} \tag{7}$$

This is referred to as the flux Jacobian. The eigensystem for C [see Eq. (1)] is

$$X^{-1} C X = B(\lambda) \tag{8}$$

which leads to the relation

$$\frac{1}{2c} \begin{bmatrix} c-u & 1 \\ c+u & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix} \begin{bmatrix} 1 & 1 \\ c+u & u-c \end{bmatrix}$$

$$= \begin{bmatrix} u+c & 0 \\ 0 & u-c \end{bmatrix} \tag{9}$$

Plus-Minus Splitting

One popular approach, which we refer to as the plus-minus (±) concept and which is used both on the conservative[14] and the nonconservative[12,13] form of the equations, is to partition the set of eigenvectors of C into two groups, one composed of those associated with the positive eigenvalues, and the other with those associated with the negative ones. These are used with the relation

$$XB(\lambda)X^{-1}=C \tag{10}$$

to reassemble new coefficient matrices, which are referred to as C^+ and C^-. For example, if λ_1 and λ_2 are positive and λ_3 and λ_4 are negative

$$C^+ = \begin{bmatrix} x_{11} & x_{12} & 0 & 0 \\ x_{21} & x_{22} & 0 & 0 \\ x_{31} & x_{32} & 0 & 0 \\ x_{41} & x_{42} & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \lambda_3 & \\ & & & \lambda_4 \end{bmatrix}$$

$$\times \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

The same result is found if the X and X^{-1} are kept dense and λ_3 and λ_4 are set equal to zero.

An example for conservation-law forms can be presented using Eq. (7). Consider subsonic flows for which $u+c$ is positive and $u-c$ is negative. One can form from Eq. (9)

$$C^+ = \begin{bmatrix} 1 & 1 \\ c+u & u-c \end{bmatrix} \begin{bmatrix} u+c & 0 \\ 0 & 0 \end{bmatrix} \frac{1}{2c} \begin{bmatrix} c-u & 1 \\ c+u & -1 \end{bmatrix}$$

$$= \frac{c+u}{2c} \begin{bmatrix} c-u & 1 \\ c^2-u^2 & c+u \end{bmatrix} \tag{12a}$$

and

$$C^- = \frac{u-c}{2c} \begin{bmatrix} u+c & -1 \\ u^2-c^2 & c-u \end{bmatrix} \tag{12b}$$

One can now construct the (±) flux vectors by forming $F^+ = C^+Q$ and $F^- = C^-Q$. The new flux Jacobians are still homogeneous of degree one but their Jacobians are not equal to C^+ and C^-. However, for the Euler equations at least, the eigenvalues of $\partial F^+/\partial Q$ and $\partial F^-/\partial Q$ are still positive and negative, respectively. (For a further discussion see Ref. 14.)

It is useful for analyzing the methods to express the C^+ and C^- matrices in another form. Define the vector $\lambda(a,b)$ element-by-element such that

$$\bar{\lambda}_m(a,b) \equiv \max\{a(|\lambda_m|-b),\ 0\} \tag{13a}$$

and the coefficient matrix $\bar{C}$ such that

$$\bar{C} = XB(\bar{\lambda})X^{-1} \tag{13b}$$

Then one can show that [using $\bar{\lambda}(1,0)$]

$$C^+ = \tfrac{1}{2}X[B(\lambda)+B(\bar{\lambda})]X^{-1} = \tfrac{1}{2}[C+\bar{C}]$$

$$C^- = \tfrac{1}{2}X[B(\lambda)-B(\bar{\lambda})]X^{-1} = \tfrac{1}{2}[C-\bar{C}] \tag{14a}$$

and retrieve the (±) flux vectors by

$$F^+ = \tfrac{1}{2}[C+\bar{C}]Q = \tfrac{1}{2}F + \tfrac{1}{2}\bar{C}Q, \quad F^- = \tfrac{1}{2}F - \tfrac{1}{2}\bar{C}Q \tag{14b}$$

One purpose of splitting in this fashion is to allow for the use of stable "upwind" differencing schemes for all vectors in the flux Jacobian. We know that

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{15a}$$

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0 \tag{15b}$$

represents right- and left-going wave systems, and that

$$\frac{\mathrm{d}(u)}{\mathrm{d}t} + \frac{1}{\Delta x}[B(-1,1,0)(u) + (bc)] = 0 \tag{16a}$$

$$\frac{\mathrm{d}(u)}{\mathrm{d}t} + \frac{1}{\Delta x}[B(0,-1,1)(u) + (bc)] = 0 \tag{16b}$$

are first-order stable differencing schemes for them. A possible implicit scheme[14] is given by

$$[I + hB(-1;1;0)(C^+) + hB(0;-1;1)(C^-)](\Delta Q)$$
$$= -h/2[B(1;-4;3;0;0)(F^+) + B(0;0;-3;4;-1)(F^-)](Q) \tag{17}$$

where $h \equiv \Delta t/\Delta x$ and $\Delta Q = Q_{n+1} - Q_n$ and all flux and matrix elements are evaluated at time level n. Here (Z) designates a space vector [see Eq. (2)]. Equation (17) is first-order accurate (spatial) on the left (implicit) side and second-order accurate on the right (explicit) side. If we introduce the definitions given in Eqs. (13) and (14) into Eq. (17), we can show that Eq. (17) is equivalent to

$$[I + \tfrac{1}{2}hB(-1;0;1)(C) - \tfrac{1}{2}hB(1;-2;1)(\bar{C})](\Delta Q)$$
$$= -\tfrac{1}{4}h[B(1;-4;0;4;-1)(F) + B(1;-4;6;-4;1)(\bar{C}Q)] \tag{18}$$

which is still first-order and second-order accurate (spatial) on the left and right sides, respectively; however, now all of the terms involve central differencing. The banded matrices with nonzero central diagonals are dissipative and, since $\bar{C} \neq 0$, the dissipation persists throughout the field. It appears as a fourth derivative term on the right side and a second derivative term on the left. Compare this with the more conventional factored implicit scheme used, for example, by Pulliam and Steger.[17] The latter scheme uses central differences for all terms and deliberately adds second-order dissipation to the implicit side and fourth-order dissipation to the right side. It is similar to Eq. (18) if ϵ_I and ϵ_E were matrices formed as $\epsilon_I = 0.5h\bar{C}$ and $\epsilon_E = 0.25h\bar{C}$. At the present writing the effect of these dissipation terms on the physics is somewhat controversial. Here we are only pointing out some similarities in methods.

MacCormack's Method

MacCormack[16] uses the local eigensystem in another way; his method does, however, have certain similarities to the preceding. MacCormack's method is predictor-corrector sequence, and for Eq. (5) it can be written in periodic-matrix-operator form (see the section on notation) as

$$[I - hB_p(0;-1;1)(\bar{C})]\tilde{Q}_{n+1}$$
$$= [I - hB_p(0;-1;1)(\bar{C}+C)](Q_n)$$

$$[I+hB_p(-1;1;0)(\bar{C})](2Q_{n+1}-Q_n)$$
$$=[I+hB_p(-1;1;0)(\bar{C}-C)](\tilde{Q}_{n+1}) \tag{19}$$

where $h\equiv\Delta t/\Delta x$ and where $\bar{C}$ is defined by Eqs. (13) using $\bar{\lambda}(½,h)$. Make the definitions

$$M_1=hB_p(0;-1;1)(C),\quad \bar{M}_1=[I-hB_p(0;-1;1)(\bar{C})]$$
$$M_2=hB_p(-1;1;0)(C),\quad \bar{M}_2=[I+hB_p(-1;1;0)(\bar{C})] \tag{20}$$

and rewrite Eq. (19) as

$$\bar{M}_1(\tilde{Q}_{n+1})=[\bar{M}_1-M_1](Q_n)$$
$$\bar{M}_2(2Q_{n+1}-Q_n)=[\bar{M}_2-M_2](\tilde{Q}_{n+1}) \tag{21}$$

Since periodic matrices commute, these can be reduced to

$$\bar{M}_2\bar{M}_1(Q_{n+1})=[\bar{M}_2\bar{M}_1-½(\bar{M}_2M_1+\bar{M}_1M_2)$$
$$+½M_2M_1](Q_n) \tag{22}$$

where, if C and $\bar{C}$ are independent of x,

$$\bar{M}_1\bar{M}_2=I-hB_p(1;-2;1)(\bar{C}+h\bar{C}^2)$$
$$½(\bar{M}_2M_1+\bar{M}_1M_2)=-½hB_p(-1;0;1)(C)$$
$$½M_2M_1=½h^2B_p(1;-2;1)(C^2) \tag{23}$$

Since C and $\bar{C}$ have a common set of eigenvectors, they commute, and use has been made of the fact that $\bar{C}C-C\bar{C}=0$.

The original fully explicit MacCormack scheme[3] can also be analyzed by the described procedure. If this is carried out one arrives at the relation

$$(\Delta Q)=[-½hB_p(-1;0;1)(C)$$
$$+½h^2B_p(1;-2;1)(C^2)](Q_n) \tag{24}$$

where again $\Delta Q=Q_{n+1}-Q_n$, and all other time evaluations are made at n. One can show that Eq. (22) reduces to

$$[I-hB_p(1;-2;1)(\bar{C}+h\bar{C}^2)](\Delta Q)$$
$$=\text{terms in MacCormack's original scheme} \tag{25}$$

If the Courant number $|\lambda_i|\Delta t/\Delta x$ is less than one for all of the characteristic speeds, that is, for all λ_i, then $\bar{C}=0$ and the method reverts to MacCormack's original explicit scheme. This scheme, like all Lax-Wendroff variations, has fourth derivative dissipation for linear systems. Thus the (±) scheme, represented by Eq. (18), and the MacCormack scheme, represented by Eq. (25), both dissipate by linear analysis according to a fourth derivative term on the right side. If solutions to these equations go to a steady state, then $(\Delta Q)=0$, and the right side represents the steady-state solution.

If the $\bar{C}$ in MacCormack's method is not zero, one of the characteristic Courant numbers is greater than one and the implicit operator is no longer the identity matrix. The additional terms include the dissipative second derivative operator $B(1;-2;1)$. Notice that the same operator appears in the analysis of the (±) method given by Eq. (18).

Relaxation and Multigrid Techniques

In the past few years considerable effort has gone into the development and application of the concept referred to as multigridding. This term is used to designate a process that carries a residual through a series of related matrices in order to accelerate the removal of the error from the initially guessed solution. The order of the matrices is related to the grid size, which becomes increasingly coarse during one part of the iteration and increasingly fine during the other. Applications to difference equations having eigenvalues that are real, or that are dominated by the real part, and are all of one sign (such as the finite-difference form of the Laplacian) have been extremely successful.[18,19] Several efforts are under way to extend the concept to numerical forms of the Euler (or hyperbolic) equations. No outstanding successes have been reported as yet, but it is anticipated that this will become a very useful approach for finding nontransient solutions to the Euler and Navier-Stokes equations in the next few years.

In studying the fundamentals of the multigrid process, some interesting aspects of classical relaxation methods have surfaced and are worth reporting (see also Ref. 20). To convey the ideas as simply as possible, we chose the following form for the equation to be solved

$$A_bu-f_b=0 \tag{26}$$

where the solution is $u=A_b^{-1}f_b$. In this section, A_b and its associated A are large sparse matrices based on the space differencing of some partial differential equations, f_b contains boundary conditions, and we assume that A_b^{-1} exists. The usual way to apply a relaxation scheme is to multiply Eq. (26) by some nonsingular conditioning matrix H and to proceed with an iteration scheme. Many popular methods can be identified with the solution of the time-dependent equation

$$u'\equiv\frac{\mathrm{d}u}{\mathrm{d}t}=H[A_bu-f_b]\equiv Au-f \tag{27}$$

For example, if

$$\begin{aligned}
&A_b=B(1,-2,1), && \text{One-dimensional operator}\\
&H=B^{-1}\left(-\beta,\frac{2}{\omega},0\right), && \text{Condition matrix}\\
&u_{n+1}=u_n+hu_n', && \text{Explicit Euler method}
\end{aligned} \tag{28}$$

the following methods are represented when $h=1$

β	ω	Method
0	1	Point-Jacobi
1	1	Gauss-Seidel
1	$2\Big/\left[1+\sin\left(\frac{\pi}{M+1}\right)\right]$	Optimum SOR (29)

where M represents the number of points in the mesh, excluding boundaries.

Eigenvalue and Eigenvector Identification

The A matrix for the point-Jacobi process applied to the one-dimensional Laplacian is $B(0.5,-1,0.5)$. This has a complete set of linearly independent eigenvectors and their corresponding eigenvalues given by

$$x_m=(x_j)_m=\sin\left[j\left(\frac{m\pi}{M+1}\right)\right]$$

$$\lambda_m=-1+\cos\left(\frac{m\pi}{M+1}\right)\qquad j\text{ and }m=1,2,\ldots,M \tag{30}$$

where the order of the matrix is M. For example if $M=5$

$$\begin{array}{rc}
 & \text{Columns of eigenvectors} \\
 & \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\
X= & \begin{bmatrix} 1/2 & \sqrt{3}/2 & 1 & \sqrt{3}/2 & 1/2 \\ \sqrt{3}/2 & \sqrt{3}/2 & 0 & -\sqrt{3}/2 & -\sqrt{3}/2 \\ 1 & 0 & -1 & 0 & 1 \\ \sqrt{3}/2 & -\sqrt{3}/2 & 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & -\sqrt{3}/2 & 1 & -\sqrt{3}/2 & 1/2 \end{bmatrix} \\
\text{Corresponding eigenvalues}\rightarrow & \begin{array}{ccccc} -0.134.. & -0.5 & -1.0 & -1.5 & -1.866.. \end{array}
\end{array} \tag{31}$$

The A matrix for the Gauss-Seidel method applied to the one-dimensional Laplacian is quite different; it is given by $B^{-1}(-1,2,0)B(1,-2,1)$ (the matrix is developed in Ref. 20). The point of interest here is that its eigenvector set is not complete. If M is odd there are $(M+1)/2$ distinct real eigenvalues, with corresponding linearly independent real eigenvectors, and there are $(M-1)/2$ defective eigenvalues, all equal, with corresponding real principal vectors. Hence the matrix cannot be diagonalized and its Jordan form for $M=5$ is

$$J=\begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & 1 & \\ & & & \lambda_3 & 1 \\ & & & & \lambda_3 \end{bmatrix} \tag{32}$$

The effect of this form on a relaxation process is brought out in the next section.

Eigenvector Annihilation

The solution of Eq. (27) using the explicit Euler method can be written

$$u_n=c_1(1+\lambda_1 h)^n x_1+c_2(1+\lambda_2 h)^n x_2$$
$$+\ldots+c_M(1+\lambda_M h)^n x_M+u_\infty \tag{33}$$

if the value of h is the same for every iteration, and the eigenvectors are linearly independent. The values of c_m depend on the initial guess and u_∞ is the correct solution. For nonstationary methods, h changes with each iteration and the process becomes

$$u_n=\underbrace{\sum_{m=1}^{M} c_m \prod_{n=1}^{N}(1+\lambda_m h_n)x_m}_{\text{transient or error}}+u_\infty \tag{34}$$

where the terms involving the eigenvectors are referred to as the transients in time-accurate applications, and as the errors in relaxation processes. Here we refer to these terms as the eigenvector decomposition of the error.

We are considering here only linear equations so if in any iteration $h_n=-1/\lambda_m$ the entire content of the eigenvector x_m is theoretically removed from the error content from that point on. Therefore, one can (theoretically) remove all of the error from the initial guess in M interactions. One is sometimes led to believe that the Gauss-Seidel method has some advantages because it generates only $M/2$ eigenvalues. Although this is true, the representation of its error content is considerably more complicated than the form given in Eq. (34) (see Ref. 20) and it still requires M iterations for total annihilation of all the principal vectors and eigenvectors that compose its exact solution.

The qualification "theoretically" was used in the previous paragraph because it is well known that the process of total annihilation (which is in principal a direct solution) is usually impractical. This is because the removal of the eigenvectors associated with the eigenvalues smallest in modulus greatly amplifies the eigenvectors associated with the eigenvalues largest in modulus, and the total procedure is numerically unstable. Fortunately, there are much better strategies than total annihilation. One of these, which is completely explicit, is the Richardson, or nonstationary, point-Jacobi method; it is very well known and is described in many texts. A form of it is discussed next.

Suppose the eigenvalues of a method are all real, negative, and fall in the interval between 0 and $-L$. This is the case for the eigensystem given in Eqs. (30) and (31) when $L=2$. Then a Richardson scheme using three-point Jacobi sweeps with

$$h_1=1/L,\quad h_2=1/(\tfrac{3}{4}L),\quad h_3=1/(\tfrac{1}{2}L) \tag{35}$$

reduces the magnitudes of all the eigenvectors associated with eigenvalues between $-\tfrac{1}{2}L$ and $-L$ to less than about 1% of their starting values. [The values of h given by Eqs. (35) are nearly as good as the optimum Chebyshev roots for the model problem and are a good compromise when the matrix only approximates the model problem.] In the case of the one-dimensional Laplacian, whose eigensystem is given in Eqs. (30), this range contains one half of the entire vector set into which the total error is decomposed. Such a property is still true (for large M) when applied to the Laplacian in two and three space dimensions. An important point of this particular strategy is that the magnitude of none of the eigenvectors was amplified in the completely explicit procedure.

Of course one can solve Eq. (27) by some implicit method or by some mixture of implicit and explicit methods, in which case the numerical stability and the arithmetic complexity of the process can change drastically. In studying implicit forms to be used on problems having two or three space dimensions, care must be taken to make the analysis on model equations that have the same number of dimensions as the problem of interest. Implicit methods are not considered here because that concept is quite well known. Jameson has used the multigrid concept with a form of the implicit ADI techniques to solve two-dimensional nonlinear transonic flow problems. It is one of the most successful relaxation schemes that has been published in this area. It should be noted that in the model linear case, the eigenvector set for ADI is the same as that for point-Jacobi.

Modifying the Eigenvalue

Eigenvector Association

It is easy to verify that the eigenvectors of $B(1,b,1)$ are independent of b, but that the eigenvalues do depend on b and are, in fact, given by

$$\lambda_m=b+2\cos\left(\frac{m\pi}{M+1}\right)\qquad m=1,2,\ldots,M \tag{36}$$

It can be shown that when two matrices admit the same set of eigenvectors, the matrix of their product also has that set, and the eigenvalues of the product matrix are the product of the eigenvalues of the individual matrices. This means that the eigensystem for

$$A=\prod_{k=1}^{K} B(1,b_k,1) \tag{37}$$

is

$$x_m = \sin\left[j\left(\frac{m\pi}{M+1}\right)\right]$$

$$\lambda_m = \prod_{k=1}^{K}\left[b_k + \cos\left(\frac{m\pi}{M+1}\right)\right] \quad j \text{ and } m=1,2,\ldots,M \qquad (38)$$

A simple application of this concept is to precondition $b(0.5,-1,0.5)$ with the three-point summing process $B(1,1,1)$ applied twice. A single point-Jacobi sweep is then represented by

$$u_{n+1} = u_n + h_n[B(1,1,1)]^2[B(0.5,-1,0.5)u_n - f] \qquad (39)$$

It is easy to show that the eigenvalues of the matrix $B(1,1,1)^2B(0.5,-1,0.5)$ are

$$\lambda_m = -1 + \cos\left(\frac{3m\pi}{M+1}\right) \qquad m=1,2,\ldots,M \qquad (40)$$

Three point-Jacobi sweeps, using the h_n in Eq. (35), would again suppress all of the eigenvectors associated with the eigenvalues in the range between $-\frac{1}{2}L$ and $-L$, but this is a set of vectors quite different from that removed by applying the Richardson process to the unconditioned Laplacian. To illustrate the point, let M be a large number and consider the eigenvectors in Eqs. (30) or (38) to be listed in order from $m=1$ to $m=M$. The diagram presented in Fig. 1 shows that about 5/6 of the error vectors that can be excited by an initial guess are reduced to less than about 1% of their starting values by the two processes applied in series. This example is given only to show that the eigensystem of relaxation processes can be manipulated in many ways to possible advantage. It is not suggested that the example is in any sense optimum, but it is important to emphasize that at no part in the explicit process was any eigenvector amplified.

Eigenvalue and Eigenvector Identification with Space Frequencies

Consider the eigenvalues λ_m of A in Eq. (27). It is often assumed that an eigenvalue with large modulus is associated with a high-frequency component in some functional decomposition of the vector u when the elements of u are ordered according to a continuous trace of u through space. (We shall refer to this as a space-continuous u vector.) In fact, it is often stated that "it is easy to remove the high-frequency waves." It is indeed easy to remove the eigenvectors of A associated with the eigenvalues of high modulus, but these are not necessarily related to high space frequencies, or, when they are, the relationship can be quite subtle.

At this point the term space frequency has not been defined clearly. In the case of Laplace's equation with Dirichlet boundary conditions the "natural" frequencies would be those appearing in the sine waves given in Eq. (38). For other equations and different boundary conditions the sine waves in Eq. (38) would not be the appropriate foundation. The principal reason for introducing the concept at all comes about in studying multigrid techniques. Certain multigrid methods combine partial eigenvector annihilation on specific meshes with interpolation processes from mesh to mesh. In forming the interpolations, functions that interpolate accurately with a low-order interpolation formula are referred to as low-frequency functions; functions that interpolate inaccurately are referred to as high-frequency functions. The implications will be made clearer in the following discussion.

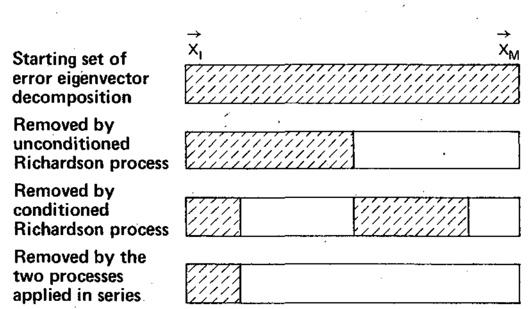

$\vec{x}_1$
$\vec{x}_M$
Starting set of error eigenvector decomposition
Removed by unconditioned Richardson process
Removed by conditioned Richardson process
Removed by the two processes applied in series

Fig. 1 Schema of error reduction without amplification.

The reason why large λ_m usually are associated with high space frequencies, and vice versa, comes about from both intuition and experience. We know that A_b in Eq. (27) is not an arbitrary matrix but that, in fact, it has some very special properties. We know that it is derived by approximating a space derivative by a matrix operation on a space-continuous vector. For example, $(\partial^2/\partial x^2)\sin mx = -m^2\sin mx$, and our intuition tells us that m^2 should relate to the eigenvalues of the A_b which has replaced $(\partial^2/\partial x^2)$ and $\sin mx$ is an eigenvector that clearly is related to a space frequency. Furthermore, most conventional methods used to relax the Laplacian numerical operators with either Dirichlet or Neumann boundary conditions form both A_b and A matrices that have the assumed property.

The precise relation between the eigenvalues and the space frequencies lies in the relation between the eigenvalues and the eigenvectors followed by the eigenvector identification with space frequency. An example of this relationship is given by Eq. (34). Remember that permutations of A_b and A, which would reorder the eigenvector elements without changing the associated eigenvalues, are permitted only if we account for the fact that the u vector is no longer space continuous. However, preconditioning of A_b to form A is acceptable without such qualification.

Mathematically, of course, the relation between eigenvalue magnitude and eigenvector "frequency" is completely arbitrary. For example, $B(1,2,1)$ has exactly the same eigenvector set as $B(1,-2,1)$, yet the relation between the $|\lambda_m|$ and their associated x_m is exactly the opposite. In the case of $B(1,2,1)$ for $M=5$ the largest $|\lambda_m|$ equals 1.866⋯ and it is associated with x_1 in Eq. (31); this vector has the lowest space frequency. On the other hand, the lowest $|\lambda_m|$ equals 0.134⋯ and it is associated with x_5, which has the highest space frequency. It is true that $B(1,2,1)$ is not related to a derivative approximation; however, this only removes it as a possibility for A_b in Eq. (27). It is still an excellent candidate for A if an inexpensive preconditioning could generate it. The preconditioning matrix

$$H = [B(1,1,1)]^2 \qquad (41)$$

used in the previous section is a step in that direction.

Consider next a matrix operator A_b that is generated by approximating a first derivative with a central difference and imposing Dirichlet boundary conditions on the left and no conditions (floating condition fulfilled by inward differencing) on the right.

$$A_b = \begin{bmatrix} 0 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & \ddots & & 1 & \\ & & & -1 & 0 & 1 \\ & & & & -2 & 2 \end{bmatrix} \qquad (42)$$

The complex eigenvalues and the first, second, and fifteenth complex eigenvectors are given in Table 1 for $M=15$. None of the eigenvectors is smooth in the real part, the imaginary part, or in the modulus. Certainly no standard interpolation procedure for the data in this form would be

Table 1 Eigenvalues and corresponding eigenvectors number 1, 2, and 15 for matrix in Eq. (42), $x=j\pi/16$

	Eigenvalues		
Number	Real	Imaginary	Modulus
1	0.33364	0.00000	0.33364
2	0.29466	0.31370	0.43039
3	0.29466	−0.31370	0.43039
4	0.21684	0.66467	0.69915
5	0.21684	−0.66467	0.69915
6	0.14737	1.02289	1.03346
7	0.14737	−1.02289	1.03346
8	0.09317	−1.35016	1.35337
9	0.09317	1.35016	1.35337
10	0.05217	−1.62394	1.62478
11	0.05217	1.62394	1.62478
12	0.02317	−1.82953	1.82968
13	0.02317	1.82953	1.82968
14	0.00580	−1.95689	1.95689
15	0.00580	1.95689	1.95689

	Eigenvectors					
	1		2		15	
j	Real	Imaginary	Real	Imaginary	Real	Imaginary
1	0.08921	0.00000	0.08139	0.07259	0.07438	0.00061
2	0.02976	0.00000	0.00121	0.04692	−0.00076	0.14555
3	0.09914	0.00000	0.06702	0.08680	−0.21045	−0.00003
4	0.06284	0.00000	−0.00627	0.09352	−0.00192	−0.26628
5	0.12010	0.00000	0.03584	0.11239	0.31061	−0.00532
6	0.10291	0.00000	−0.03097	0.13788	0.01030	0.34153
7	0.15444	0.00000	−0.01654	0.14330	−0.35765	0.01682
8	0.15444	0.00000	−0.08079	0.17492	−0.02468	−0.35826
9	0.20596	0.00000	−0.09522	0.16950	0.34328	−0.03355
10	0.22316	0.00000	−0.16202	0.19499	0.04297	0.31330
11	0.28042	0.00000	−0.20413	0.17613	−0.26956	0.05234
12	0.31671	0.00000	−0.27742	0.18285	−0.06103	−0.21390
13	0.38609	0.00000	−0.34324	0.14298	0.14866	−0.06832
14	0.44553	0.00000	−0.42342	0.11731	0.07353	0.07662
15	0.53474	0.00000	−0.50480	0.04472	−0.00085	0.07601

appropriate. However, a permutation and partial reflection of the eigenvectors [(and a corresponding permutation and partial reflection of the u vector (see Ref. 20)] does generate an eigensystem such that increasing $|\lambda_m|$ corresponds to increasing space frequencies (using the interpolation definition given earlier) in the resulting vectors and the eigenvectors associated with the lower $|\lambda_m|$ are "smooth." This result is important when multigrid concepts are applied to convective processes that are approximated by central differencing and a floating boundary condition.

An Approach to Eigenvector Annihilation through Multigridding

Consider the relaxation process given by Eq. (27) in which the eigenvalues of A are real and negative. If we define a residual vector r and an error vector e by

$$r=-Au+f \qquad e=u_\infty-u \tag{43}$$

then one can show that

$$Ae-r=0 \tag{44}$$

which defines the problem on the fine grid. If we use the Richardson three-step process on Eq. (27) with the three step sizes given in Eq. (35), at the end of the three steps

$$e=\sum_{m=1}^{M/2} c_m x_m \prod_{n=1}^{3}(1+\lambda_m h_n) + \underbrace{\sum_{m=M/2+1}^{M} c_m x_m \prod_{n=1}^{3}(1+\lambda_m h_n)}_{\text{Amplitudes} \le 0.01 \text{ times initial value}} \tag{45}$$

Consider the error content in the x_m vectors from $m=(M/2)+1$ to M to be "removed." Since this amounts to half of the information content in the data, we assume the remaining eigenvectors carry redundant information. In other words, we assume we can permute the contents of e into two vectors e_a and e_b, each about half the length of e, such that there is an approximating relation that will express e_b in terms of e_a. Under these conditions we can write

$$\begin{bmatrix} e_a \\ e_b \end{bmatrix} = Pe \tag{46}$$

so that

$$PAP^T Pe = Pr \tag{47}$$

becomes

$$\left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right] \begin{bmatrix} e_a \\ e_b \end{bmatrix} = \begin{bmatrix} r_a \\ r_b \end{bmatrix} \tag{48}$$

and without approximation we can write

$$A_1 e_a + A_2 e_b = r_a \tag{49}$$

The approximating relation

$$e_b \approx A_2' e_a \tag{50}$$

can now be introduced and there results

$$[A_1 + A_2 A_2'] e_a = r_a \tag{51}$$

which defines the problem on the coarser grid.

In the cases in which increasing $|\lambda_m|$ corresponds to increasing space frequencies in the eigenvectors, the two vectors can be chosen such that e_a is composed of the data on the even points, e_b contains that on the odd points, P is an odd-even permutation, and A_2' represents some standard interpolation procedure including the boundaries. Variations of this technique have been used to accelerate relaxation processes.[21]

Zonal Finite-Difference Methods for the Euler Equations

Grid Topologies and Data Bases

Let us consider the problem of computing a flowfield about a realistic aerodynamic body. In two dimensions this would be, for example, an airfoil, an airfoil in a cascade, or multiple airfoils. In three dimensions it might be a wing, a wing mounted on a body, or a wing-body-nacelle combination. For finite-difference or finite-volume applications, most codes at present fill the boundaries and interior of the physical domain, composing the flowfields being computed, with a set of points in such a way that each point can be mapped onto one and only one point in a grid of equispaced points on and in a rectangular computational domain. Grids with this property are called topological boxes. Developing methods for generating these grids is a highly specialized field and entire conferences are devoted to their presentation.[22]

Suppose we store the value of a dependent variable at each one of these points in a computer memory and refer to this information as a data base. Next suppose we arrange the storage of this data base in such a way that the location of a word in the base completely identifies its location in the computational box and hence in physical space. Finally, suppose this arrangement is carried out such that adjacent points in memory are (except at boundaries) adjacent in the physical mesh in one of the space directions, and simple data permutations reformulate the data base in such a way that the adjacency property exists for the other two space directions. Grids and data bases that have this property are referred to as topological box grids and data bases, and the data are said to be stored in ordered sets of space-continuous vectors.

Solving the Euler Equations in Topological Boxes

The availability and use of data bases formed from topological box grids profoundly affects the coding and thought processes that are used to form finite-difference methods. Let

$$\frac{\partial Q}{\partial \tau} + \sum_{i=1}^{3} \frac{\partial F_i}{\partial \xi_i} = 0 \qquad (52)$$

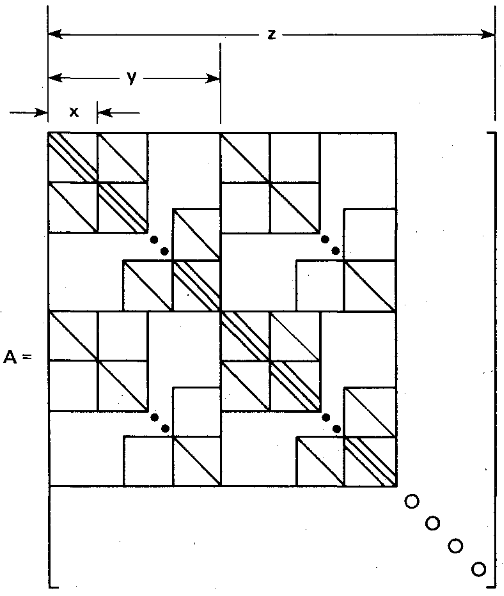
represent the Euler equations in three-dimensional conservation-law form with generalized coordinates $\xi_i = \xi_i(x,y,z,t)$. The usual definitions hold for the vector Q and the flux vectors F. If one forms the flux Jacobians as in Eq. (7), using C to designate $\partial F/\partial Q$,

$$\frac{\mathrm{d}Q}{\mathrm{d}\tau} + \sum_{i=1}^{3} \delta_{\xi_i} C_i Q = 0 \qquad (53)$$

where δ_ξ is a difference operator that approximates a first derivative and operates on the product CQ. Equation (53) represents a large set of coupled nonlinear ordinary differential equations. The matrix-vector relation formed by this set is written

$$\frac{\mathrm{d}(Q)}{\mathrm{d}\tau} = A(Q) + (bc) \qquad (54)$$

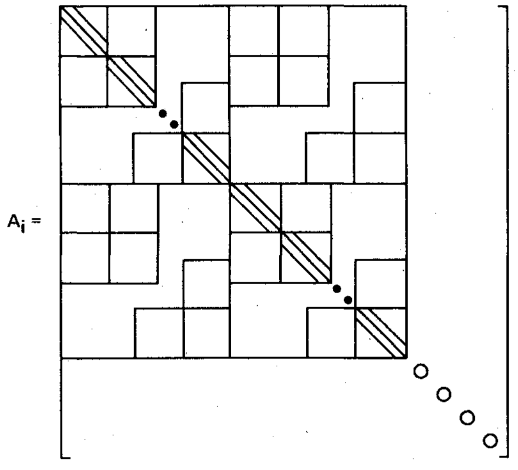
where (bc) is a space vector containing the boundary conditions and (Q) is a space vector of the five-element vector Q. If (Q) is space continuous and δ_ξ is based on a three-point central difference scheme, the matrix A has the form shown in Eq. (55).



(55)

All elements of the matrix are zero except along the diagonal bands indicated by lines. The diagonal bands represent locations where dense 5×5 matrices may be stored. Without damping, the central diagonal would also be zero except at floating boundaries.

The form of A displayed in Eq. (55) is the "signature" of a finite-difference scheme applied to a space domain filled with a grid that forms a topological box. This form is especially useful when factored implicit methods are used to solve Eq. (53) inside the domain. A well-known factorization[5] is carried out by choosing an implicit time-marching differencing scheme and then factoring the left side into three matrices A_i, $i = 1,2,3$, one for each space coordinate,



(56)

such that each A_i has the form shown in Eq. (56). Solutions of the implicit algorithm then amount to applying simple block-tridiagonal solvers to the data base in the three permutations that align it into space-continuous vectors.

Zonal Method Prospects

Solving flowfields by using finite-difference or finite-volume methods with data bases that are produced from box topologies can be made into a very efficient and accurate process for the points involved. Unfortunately, there are two major drawbacks, both of which apply most strongly in three dimensions. One is the difficulty of mapping complicated geometries into a single box. The other is the possible waste of computing time and memory locations that can be caused by having to extend lines of points needed for local clustering across the entire width of the box and into regions where they are unnecessary.

In finite-difference applications a foreseeable trend that may be practical for three-dimensional geometries is the development of flexible zonal methods. A zonal method is one that subdivides the total flow region into interconnected smaller regions, each one of which is a topological box. Let A be a matrix of the form shown in Eq. (55), and let L and U be sparse matrices with no particular banded structure. Then the solution process of a zonal method is schematically represented by Eq. (57) in which A is replaced by $\mathbf{A}$ where

$$\mathbf{A} = \left[\begin{array}{c|c|c} A & U & U \\ \hline L & A & U \\ \hline L & L & A \end{array}\right] \qquad (57)$$

for a three-zone application. The lower interface blocks L and the upper interface blocks U are all different and probably will be treated explicitly in the time-march process, but the diagonal A blocks can be individually advanced by implicit techniques. Such methods should, on the basis of physical arguments, be stable within a Courant number of one based on the zone size, and they should be highly competitive with finite-element methods, which appear at this time to be the most attractive alternative, principally because of their geometric flexibility.

Some versions of zonal methods in three dimensions are being attempted, and preliminary results are reported in the papers by Forsey et al. and by Lee et al. in Ref. 22. In those papers the technique is referred to as grid patching and multiblock grid construction, respectively. If the theoretical maximum zonal Courant number of one can be realized in practice, the principal problem will be to maintain accuracy at the interface under the condition that the interface be highly adaptable. In such a case the grids that are interior to the individual zones can be constructed by any of the standard ways, and standard methods can be used to compute the solutions there. Progress along these lines will be a real challenge for the 1980's.

Computing High Reynolds Number Flowfields

During the 1970's many computer codes were written to study two-dimensional flowfields that had shock-wave boundary-layer interaction and various kinds of flow separation phenomena. Of interest here are the kind that had turbulent boundary layers. (For a discussion and list of references see Ref. 23.) A few codes were used to study three-dimensional flows of the same type, but computer resources in the 1970's caused this class to be rather limited. For the most part, each code applied to a specific experiment and adaptation to other experiments was difficult at best. These studies demonstrated that with a variety of methods it is possible to write computer programs that will calculate pressure distributions about various aerodynamic shapes in flowfields having shock waves and small regions of separation, and that these pressure distributions are accurate enough for many engineering applications. Most of the programs were constructed to simulate a particular experiment and were made to work well for that experiment. In each case this demonstrated the possible capability of the programs, but usually the same program was not applied to a variety of experiments, in which case reliability was impossible to assess.

It is hoped that measuring the reliability of codes that can be used to approximate separated turbulent flows will be a goal for the 1980's. Achievement of this goal will be approached if the same codes will give satisfactory results for a variety of nonsimilar experiments. This will be touched upon again at the end of this section.

Table 2 Flows that can and cannot be computed without some modeling of subgrid scales

	Order of |eigenvalue|	Eigenvector	Scale definition
Flows governed by linear equations	1	x_1	"Low"
	2	x_2	
	3	x_3	
Low Reynolds number laminar flows	4	x_4	Computational scale
	5	x_5	
Unconfined low Reynolds number turbulence	⋮	⋮	
	$M-1$	x_{M-1}	
	M	x_M	"High"
Euler equations with captured shocks High Reynolds number laminar flows General turbulence		Effect must be modeled	Subgrid scale migration

Basic Equations

The equations basic to the computation of flows that have shock waves, turbulent boundary layers, and regions of separation are really not uniquely defined. Using the left side of Eq. (52) to define the Euler terms, the equations can be expressed as

$$\begin{array}{l}\text{Euler} \\ \text{terms}\end{array} + \begin{array}{l}\text{Approximation for} \\ \text{subgrid scales}\end{array} = \begin{array}{l}\text{Molecular viscosity} \\ \text{terms}\end{array} \qquad (58)$$

The crucial issue is the implementation of an approximation for the effects of subgrid phenomena. This is divided into two parts

$$\begin{array}{c}\text{Approximation for} \\ \text{subgrid scales}\end{array} \left\{ \begin{array}{l} \text{1) High-frequency dissipation} \\ \text{2) Turbulence modeling} \end{array} \right. \qquad (59)$$

High-Frequency Dissipation

Some definitions of terminology are in order at this point. Consider the Euler equations and choose a discrete grid, which is used together with some numerical method to convert these equations into a set of difference equations. Starting from an initial guess, compute the time history of the flowfields. As we have seen from the derivation of Eq. (34) one can sometimes analyze this process in terms of the eigensystem belonging to the matrix formed by the space differencing. Assume for the purpose of discussion that the eigenvectors form a complete set and that their eigenvalues are distinct. One can see from Eq. (34) that we can think of this set as a set of discrete eigenfunctions or basis functions into which the solution can be decomposed. A special example occurs in use of spectral methods. Let us order the eigenvectors according to the modulus of their eigenvalues, and assume that high frequencies correlate with high-numbered eigenvectors, which would disappear from the set if the mesh size were decreased. Let there be M points in the grid. Under the assumptions stated there will be M eigenvectors in the set of basis functions, and we can construct Table 2. In linear problems eigenvectors do not interact and one set remains for all time. In nonlinear cases, such as the Euler and Navier-Stokes equations, eigenvectors of higher order constantly are being generated by nonlinear interactions of lower order ones. However, if they are damped sufficiently by the molecular viscosity, the eigenvector range will not exceed the limit that can be supported by the mesh. For problems such as these no frequencies migrate outside the available eigenvector set and no approximations at the high-frequency end are necessary.

The preceding is not true for problems involving the Euler equations with captured shocks, high Reynolds number laminar flows, and turbulence in general. In these cases, without some control of the higher frequencies that are constantly being generated, they "leave" the mesh and reappear as amplitudes of lower ones, a process called nonlinear aliasing. If aliasing is suppressed (in general a difficult job) the high frequencies "pile up" and numerical instability or simply "hash" appears, which destroys any physical reality of the solution. One wants to avoid the generation of terms that migrate out of the mesh and alias back, or that migrate to the limiting capacity of the mesh and pile up. It would be preferable to let the terms escape and to model their influence back on those that remain. Except in certain special cases (see the next subsection) this kind of model generally is not attempted. The standard approximation is to reduce the amplitude of the migrating terms by some form of numerical dissipation as they move into the high-frequency range, and to destroy them before they exceed the mesh capacity. This dissipation is the standard "model" used in most codes that require control of migration to the subgrid scales. The physical justification is simply that the high-frequency terms have small amplitudes and, therefore, whether removed or not have no important effect on the high-amplitude low-frequency terms. This gives the rationale for including some form of high-order damping in codes that are used to solve the Euler equations.

The dissipation model just described is a rather crude one. For example, the amount and type of dissipation that is used varies from method to method, and is quite arbitrary except that it should lie in the range of some truncation error. A very large literature exists on the subject pertaining to the process of "capturing" shocks in the numerical analysis of inviscid supercritical flows. Here, in effect, the dissipation models are judged by the extent to which they destroy the crispness of the shock while maintaining a stable calculation. Many algorithm developments are continuing in an attempt to improve this area.

Very little has been written about the effect of these models on calculating high Reynolds number laminar flows or flows with turbulence. In the case of turbulent flows, the dissipation models are working on the scales that have low energy and high frequency relative to the mean flow, but that have high energy and very low frequency relative to the range of true viscous dissipation. They are distinct from the turbulence modeling described next but they play an essential role in calculating the effects of turbulence.

Turbulence Modeling

There is a case in which the form of approximation used for subgrid scales is much more critical. This occurs when computing the effect of turbulent flows. It is well known that turbulence itself cannot be computed at high Reynolds numbers, simply because the scale range (or eigenvector range) is much too large for modern computer memories. For example, a grid of 512^3 in a Cartesian mesh used to support the solution of an incompressible homogeneous turbulent flow without modeling would require one billion words of memory if all possible frequency combinations were included. Even this might not establish a three-dimensional inertial range.

Present day computer memories permit about 64^3 grid points for a compressible flow approximation; this is about enough to permit a clustering of points in one direction only that will resolve the mean profile of a turbulent shear layer very near (down to $y^+ \approx 6$) a body surface. The structure of these mean profiles (steady or unsteady) is used with empirically established equations to approximate the effect of the subgrid structure back onto the mean flow. These approximations are established on the basis of the Reynolds-averaged Navier-Stokes equations and are referred to as turbulence models.

Several basic algebraic, one-equation and two-equation turbulence models have been developed and used to analyze turbulent flows with regions of separation at high Reynolds numbers (for references see Ref. 23). Most of them give quite similar results for pressure distributions along a surface, but begin to deviate significantly when used to predict more sensitive quantities such as skin friction.

One algebraic turbulence model,[24] patterned after that of Cebeci,[25] was designed in such a way that the length scales are proportioned to the width of vorticity layers. It is reasonably robust through regions of separation. It has been used without modification to simulate a variety of steady two-dimensional flows, unsteady two-dimensional flows, steady axisymmetric flows, and steady three-dimensional flows. The results have been compared with experimental pressure distributions, and it is evident that these results contain information that could be quite useful for preliminary engineering approximations.

The study described in the last paragraph should be only preliminary to what will develop in the later 1980's, when fast, high-capacity computers become much more widely available. There is good reason to believe that very useful computer codes that simulate shocks and mild forms of steady and unsteady separation in two and three dimensions can be written now without furthering the development of turbulence models. Admittedly better turbulence models would be highly desirable and work in this area should and will continue. However, for a great many problems of interest, the need for larger computer memory size, more robust numerical methods, faster convergence rates, and practical three-dimensional mesh generation will be the pacing items of the 1980's rather than the modeling of turbulence.

Scientific Languages

At present the vast majority of aerospace engineers express themselves to a computer through a FORTRAN compiler, and FORTRAN 77 is the most modern version of the language. Other languages that are available are Pascal and the recently released ADA.[26] The statement is made by computer scientists that FORTRAN is a primitive language and should be abandoned in favor of a more advanced one such as Pascal or ADA. This subject is considered here in light of the experience that has been accumulated by designers of codes for three-dimensional problems requiring large memory capacities, and also for more typical users with more moderate memory requirements who are engaged in method development and two-dimensional flow analysis. Three issues are considered: 1) how the code design time and construction time are used for the three-dimensional-type problem, 2) whether one of the languages mentioned would be the most useful in helping to reduce those times, and 3) whether those engaged in numerical research of fluid mechanics could benefit by a particular higher level language.

Experience in constructing codes that are designed with an attempt to cope efficiently with massive amounts of memory dictates that by far the most difficult and time-consuming part of the code construction is the design and management of the data base. This is subject to the constraints imposed by the architecture of each individual computer. The significant factors are considerations, such as the number of hierarchies of memory that the data must be buffered through to get to the arithmetic unit, the size of the vector strings that have to be identified for efficient operation, and the ratios of the various transfer rates to the speed of the processor. These are matters that pertain more to operating systems than languages. If the operating system cannot by itself manage the massive data transfer and vector identifications efficiently (as is the usual case for systems with or without virtual memory), the user must be prepared for a great deal of expense of some nontrivial code development. The use of predesigned database systems[27] is one possible relief for this situation but it can impose annoying constraints.

None of the difficulties are alleviated by the use of either FORTRAN or Pascal. In fact, experience has shown that most scientific codes used to study fluid mechanics are compartmentalized into subroutine structures. According to a study made by Watson[28] the use of FORTRAN or Pascal is not a major issue to those engaged in computational fluid dynamics, if the code is being constructed from its inception. A compiler is not yet available for ADA, so judgments regarding it really cannot be made as yet, but there is no strong indication that its availability will alter the situation significantly.

Concluding Remarks

Although the development of numerical techniques for analyzing compressible, viscous, and inviscid fluid flows has seen a remarkable growth over the last 30 years, it is the author's opinion that the available techniques still must be considerably improved along the lines of convergence rates, reliability, and geometric adaptability. Those investigators using finite-difference methods cannot ignore spectral and finite-element approaches, but in their own realm the use of local eigensystems, multigriding, and zonal methods appears to have attractive possibilities.

References

[1] Chapman, D. R., "Computational Aerodynamics Development and Outlook," *AIAA Journal*, Vol. 17, Dec. 1979, pp. 1293-1313.

[2] Richtmeyer, R. D. and Morton, K. W., *Difference Methods for Initial-Value Problems*, 2nd ed., Interscience Publishers, New York, 1967.

[3] MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, 1969.

[4] Briley, W. R. and McDonald, H., "Solutions of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique," *Lecture Notes in Physics*, Springer Verlag, Vol. 35, pp. 105-110.

[5] Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

[6] MacCormack, R. W., "An Efficient Explicit-Implicit Characteristic Method for Solving the Compressible Navier-Stokes Equations," *SIAM-AMS Proceedings*, Vol. 11, 1978, pp. 130-155.

[7] Lomax, H., "Recent Progress in Numerical Techniques for Flow Simulation," *AIAA Journal*, Vol. 14, April 1976, pp. 512-518.

[8] Hughes, T.J.R., ed., *Finite Element Methods for Convection Dominated Flows*, AMD-Vol. 34, ASME, New York, 1979.

[9] Orzag, S. A., "Spectral Methods for Problems in Complex Geometries," *Numerical Methods for PDE's*, Academic Press, Inc., 1979.

[10] Courant, R., Isaacson, E., and Rees, M., "On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences," *Communications on Pure and Applied Mathematics*, Vol. 5, Aug. 1952, p. 243.

[11] Steger, J. L., "Coefficient Matrices for Implicit Finite Difference Solution of the Inviscid Fluid Conservation Law Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 13, Feb. 1978, p. 175.

[12] Moretti, G., "The λ-Scheme," *Computers and Fluids*, Vol. 7, Sept. 1979, p. 191.

[13] Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "The Split Coefficient Matrix Method for Hyperbolic Systems of Gas-Dynamic Equations," AIAA Paper 80-0268, 1980.

[14] Steger, J. L. and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamics Equations with Application to Finite Difference Methods," NASA TM 78605, 1979.

[15] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow," *Journal of Computational Physics*, Vol. 23, March 1977, p. 263.

[16] MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper 81-0110, Jan. 1981.

[17] Pulliam, T. H. and Steger, J. L., "On Implicit Finite Difference Simulations of Three-Dimensional Flow," *AIAA Journal*, Vol. 18, Feb. 1980, pp. 159-167.

[18] Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, Vol. 31, April 1977, pp. 333-390.

[19] Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes to the Multigrid Method," AIAA Paper 79-1458, 1979.

[20] Lomax, H., "Eigensystem Analysis of Classical Relaxation Techniques with Application to Multigrid Methods," NASA Report (to be published).

[21] Lomax, H., Pulliam, T. H., and Jespersen, D. C., "Eigensystem Analysis Techniques for Finite-Difference Equations—I. Multi-Level Techniques," AIAA 81-1027, 1981.

[22] *Numerical Grid Generation Techniques*, Workshop Proceedings, NASA CP 2166, 1980.

[23] Mehta, U. and Lomax, H., "Reynolds-Averaged Navier-Stokes Computations of Transonic Flows—The State of the Art," presented at the Symposium on Transonic Perspective, Ames Research Center, Feb. 1981.

[24] Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, 1978.

[25] Cebeci, T., "Calculations of Compressible Turbulent Boundary Layers with Heat and Mass Transfer," *AIAA Journal*, Vol. 9, June 1971, pp. 1091-1097.

[26] Johnson, R. C., "Special Report: ADA, The Ultimate Language?" *Electronics*, Vol. 53, Feb. 10, 1981, pp. 127-132.

[27] Lomax, H. and Pulliam, T. H., "A Fully Implicit, Factored Code for Computing Three-Dimensional Flows on the ILLIAC IV," edited by G. Rodrigue, *Parallel Computations* (to be published).

[28] Watson, V., "Comparison of Pascal and FORTRAN for Solving Problems in the Physical Sciences," submitted to the AIAA Computers in Aerospace III Conference, Oct. 1981.