NASA Technical Memorandum 84310

(NASA-TM-84310) NUMERICAL COMPUTATION OF TRANSONIC FLOW GOVERNED BY THE FULL-POTENTIAL EQUATION (NASA) 114 p HC A02/MF A01 CSCL 01A

N83-19708

Unclas
G3/02 02903

Numerical Computation of Transonic Flow Governed by the Full-Potential Equation

Terry L. Holst, Ames Research Center, Moffett Field, California

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

NUMERICAL COMPUTATION OF TRANSONIC FLOW GOVERNED BY THE FULL-POTENTIAL EQUATION

Terry L. Holst

Applied Computational Aerodynamics Branch
NASA Ames Research Center, Moffett Field, California, U.S.A.

To be presented at the lecture series on Computational Fluid Dynamics,

March 7-11, 1983,

von Kármán Institute, Rhode-St-Genese, Belgium

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

2. GOVERNING EQUATIONS 2
   2.1 Classification of Second-Order Partial Differential Equations 2
   2.2 Transonic, Small-Disturbance Potential Equation 4
   2.3 Full-Potential Equation 5

3. CLASSICAL RELAXATION ALGORITHMS 8
   3.1 Notation Conventions 8
   3.2 The von Neumann Test for Stability 10
   3.3 The Point-Jacobi Scheme 10
   3.4 The Point-Gauss-Seidel Scheme 12
   3.5 The Successive Overrelaxation Scheme 12
   3.6 Block or Line Iteration Schemes 14
   3.7 The Alternating Direction Implicit Scheme 14
   3.8 Convergence Rate Estimation 16

4. ALGORITHMS FOR THE FULL-POTENTIAL EQUATION: EARLY IDEAS 16
   4.1 Subsonic Difference Scheme 17
   4.2 Simple Supersonic Difference Scheme 18
   4.3 Rotated Supersonic Difference Scheme 21
   4.4 Temporal Damping 23

5. TRANSFORMATION AND GRID-GENERATION TECHNIQUES 24
   5.1 Governing Equation Transformation 24
   5.2 The Elliptic Grid-Generation Procedure 26
   5.3 The Hyperbolic Grid-Generation Procedure 28
   5.4 The Parabolic Grid-Generation Procedure 29
   5.5 Geometric and Analytic Mapping Procedures 30
   5.6 Solutions-Adaptive Grid-Generation Schemes 31

6. SPATIAL DIFFERENCING SCHEMES 32
   6.1 The Finite-Volume Scheme 32
   6.2 The Artificial Density Scheme 34
   6.3 Spatial Differencing Schemes Based on Flux-Vector Splitting . 36
   6.4 Free-Stream Consistency Conditions 37
   6.5 Nonisentropic Full-Potential Formulation 40
   6.6 Other Spatial Differencing Schemes 42

7. ITERATION SCHEMES 43
   7.1 The Alternating Direction Implicit Scheme 43
   7.2 The AF2 Approximate Factorization Scheme 45
   7.3 Convergence Characteristics of SLOR, ADI, and AF2 Iteration Algorithms 50
   7.4 AF3 Approximation-Factorization Scheme 51
   7.5 Multigrid Iteration Schemes 52
   7.6 Other Iteration Schemes 55

8. APPLICATIONS IN THREE DIMENSIONS 57
   8.1 Computer Code Characteristics 57
   8.2 Results Obtained with Existing Codes 60

9. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH 65

Page

REFERENCES 68

TABLES 82

FIGURES 85

NUMERICAL COMPUTATION OF TRANSONIC FLOW GOVERNED BY THE FULL-POTENTIAL EQUATION

Terry L. Holst

Ames Research Center

1. INTRODUCTION

This set of notes discusses the numerical solution of transonic flow fields governed by the full-potential equation. In a general sense this presentation deals with relaxation schemes suitable for the numerical solution of elliptic partial differential equations. Of course, transonic flow is not purely elliptic in nature but consists of hyperbolic regions embedded in an otherwise elliptic domain. However, the most successful numerical methods of solution for transonic flow applications, at least for potential formulations, have evolved from classical relaxation schemes associated with elliptic equations. Thus, most of the algorithms presented herein will have an elliptic-equation, relaxation-algorithm flavor. For related material the reader is referred to Hall (ref. 1) for a review with emphasis on the historical aspects of computational fluid dynamics (CFD), Holst et al. (ref. 2) for a general review of computational transonic aerodynamics (CTA), Chapman (refs. 3,4) for a general review of current CFD research, and Kordulla (ref. 5) and Baker (ref. 6) for additional information regarding transonic-flow solution methods.

The transonic speed regime provides the most efficient aircraft cruise performance; hence, most large commercial aircraft cruise in this speed regime. However, transonic flow fields tend to be sensitive to small perturbations in flow conditions or to slight changes in geometrical characteristics, either of which can cause large variations in the flow field. Large performance penalties can result because of relatively small perturbations away from desired design conditions. Computational techniques, therefore, have enjoyed an increasing role in helping the aerodynamics engineer find optimal design conditions, as well as to evaluate design sensitivity.

Transonic flow fields contain a variety of interesting and unique characteristics. Typical airfoil and swept-wing flow fields are shown in figures 1 and 2. The outer free-stream flow is typically subsonic and elliptic in nature. Regions of supersonic flow (hyperbolic) usually exist on the upper airfoil or wing surface and are generally terminated by a weak "transonic" shock wave. For the case of a swept-wing flow field, the shock wave may actually consist of a system of shocks, as shown in figure 2. The first shock is swept and therefore has a supersonic downstream Mach number. The rear shock is approximately normal to the local flow and therefore has a subsonic downstream Mach number.

Signals tend to propagate very rapidly downstream in transonic flow fields where the propagation velocity is $u + a$ (local fluid speed plus sonic velocity) and very slowly upstream where the propagation velocity is $u - a$. For a downstream disturbance to propagate upstream it must move around the supersonic zone, further increasing the difference between the upstream and downstream propagation speeds. This situation tends to make transonic numerical solution techniques, which depend on physical-time-dependent algorithms, very slow. Such problems are said to be "stiff," and they require large amounts of computer time for even a single steady-state solution.

Another characteristic of transonic flow is that it is governed by equations that are inherently nonlinear. Linearization of these equations will remove the vital flow-field physics which are responsible for the prediction of shock waves. This nonlinear behavior tells us that a direct (i.e., noniterative) solution procedure for transonic flow is theoretically impossible. Thus, one basic feature associated with all of the schemes discussed within these notes is that they are all iterative.

Viscous effects are also important in transonic flow fields. This complex subject involves four major effects: (1) shock/boundary-layer interaction effects, (2) the decambering and thickness effects caused by the addition of a simple displacement thickness, (3) trailing-edge effects, and (4) near-wake effects. However, a discussion of viscous correction procedures is not within the scope of the present set of notes. Instead, the interested reader is referred to Lock (ref. 7) and Melnik (ref. 8) who treat this subject in detail.

The following presentation can be divided into two general categories: an introductory treatment of the basic concepts associated with the numerical solution of elliptic partial differential equations and a more advanced treatment of current procedures used to solve the full-potential equation for transonic flow fields. The introductory material is presented for completeness; it covers governing equations (sec. 2), classical relaxation schemes (sec. 3), and early concepts regarding transonic, full-potential equation algorithms (sec. 4). These topics are intended to provide an introduction for some of the more advanced concepts presented in the later sections. The more knowledgeable reader could skip sections 2-4 and proceed directly to sections 5-8 without a significant loss in continuity.

State-of-the-art topics concerning the numerical solution of the full-potential equation for transonic flows are presented in sections 5-8. These sections include a discussion of equation transformation and grid-generation procedures (sec. 5); a presentation of recent full-potential spatial differencing schemes (sec. 6); a presentation of full-potential iteration schemes, with special emphasis on convergence acceleration (sec. 7); and a brief review of recent three-dimensional applications (sec. 8). These notes are then concluded with a few general remarks including recommendations for future research.

2. GOVERNING EQUATIONS

2.1 Classification of Second-Order Partial Differential Equations

Consider the following general quasi-linear, second-order partial differential equation (PDE):

$$Au_{xx} + Bu_{xy} + Cu_{yy} = F \tag{2.1}$$

where A, B, C, and F are functions of x, y, u, u_x, u_y. This equation can be classified by considering the corresponding characteristic equation (for a derivation of the characteristic equation and additional discussion on this topic see refs. 9 and 10):

$$A\left(\frac{dy}{dx}\right)^2 - B\,\frac{dy}{dx} + C = 0 \tag{2.2}$$

Using the quadratic formula, the two characteristic directions associated with equation (2.1) are given by

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{B \pm \sqrt{B^2 - 4Ac}}{2A} \tag{2.3}$$

The nature of these characteristics determines the equation classification. Equation (2.1) is hyperbolic if the characteristics are real and distinct [that is, if the discriminant of eq. (2.3) is greater than zero ($B^2 - 4AC > 0$), then the equation is hyperbolic]; parabolic if the characteristics are real and coincidental ($B^2 - 4AC = 0$); and elliptic if the characteristics are complex and distinct ($B^2 - 4AC < 0$).

Several classical PDEs along with their classifications are given as follows.

Laplace's equation:

$$u_{xx} + u_{yy} = 0$$

$$\left(\frac{dy}{dx}\right)_{1,2} = \pm i \quad \text{elliptic}$$

Heat equation:

$$u_y = \sigma^2 u_{xx} \quad (\sigma \sim \text{real})$$

$$\left(\frac{dy}{dx}\right)_{1,2} = \pm 0 \quad \text{parabolic}$$

Wave equation:

$$u_{xx} = c^2 u_{yy} \quad (c \sim \text{real})$$

$$\left(\frac{dy}{dx}\right)_{1,2} = \pm c \quad \text{hyperbolic}$$

The primary motivation for studying the nature of PDEs in the present context is to gain insight into the physics they govern, and, therefore, to develop guidelines for the implementation of numerical solution procedures. Different equation types generally require different solution algorithms.

A particular point, P, in a solution domain governed by a PDE, has associated with it regions called domains of influence and domains of dependence. These domains are determined by characteristics. For example, in the case of steady supersonic flow, which is hyperbolic, the domain of dependence is defined by the forward Mach cone, and the domain of influence is defined by the aft Mach cone (see fig. 3). Physically, the flow conditions at point P depend on the flow conditions in the domain of dependence. Conversely, the flow conditions at point P can influence any other point in the domain of influence.

The numerical domain of dependence, that is, the domain of dependence modeled by the numerical differencing scheme, should mimic the physical domain of dependence as closely as possible. If it does not, instability probably will result.

Elliptic equations are much different in nature than hyperbolic equations. A single point in an elliptic solution domain influences every other point and vice versa; that is, the elliptic domains of dependence and influence are coincidental and contain the entire solution domain. Solution algorithms for elliptic equations should reflect this fact to properly simulate the physics.

For transonic flow applications, the solution domain contains both elliptic (subsonic) and hyperbolic (supersonic) regions. The boundaries between these regions (sonic lines or shock waves) are unknown in advance and must be determined as part of the solution. Most successful relaxation schemes used for transonic applications utilize a feature called type-dependent differencing which allows the local differencing scheme to be adjusted to the local flow-field physics. Numerical solution techniques for implementing this philosophy will be the primary subject of discussion in subsequent sections of these notes. Two PDE formulations representing different levels of approximation for transonic flow fields are discussed next.

2.2 Transonic, Small-Disturbance Potential Equation

The transonic, small-disturbance (TSD) equation expressed in two-dimensional Cartesian coordinates (x,y) is given by

$$[1 - M_\infty^2 - M_\infty^2(\gamma + 1)\phi_x]\phi_{xx} + \phi_{yy} = 0 \qquad (2.4)$$

where M_∞ is the free-stream Mach number, γ is the ratio of specific heats, and ϕ is the small-disturbance or perturbation velocity potential defined by

$$\nabla\phi = \vec{q} - \vec{q}_\infty \qquad (2.5)$$

In equation (2.5), $\vec{q}$ and $\vec{q}_\infty$ are the local and free-stream velocity vectors defined by

$$\vec{q} = u\hat{i} + v\hat{j} \ , \quad \vec{q} = U_\infty\hat{i} \qquad (2.6)$$

where $\hat{i}$ and $\hat{j}$ are the unit vectors along the x and y directions, respectively. Thus, the velocity components are given by

$$\phi_x = u - U_\infty \ , \quad \phi_y = v \qquad (2.7)$$

Boundary conditions for a typical "thin" airfoil used in conjunction with equation (2.4) are given by

$$\phi_y(x,0^\pm) = U_\infty \frac{dg^\pm}{dy} \qquad (2.8)$$

where $g^+(x)$ and $g^-(x)$ define the upper and lower airfoil surfaces, respectively. Application of this boundary condition is made at the airfoil slit (or chord line). This simulates the required flow-tangency boundary condition at the airfoil surface to an accuracy consistent with small-disturbance theory. An auxiliary relation, usually used in this formulation to define the airfoil surface pressure coefficient, is given by

$$C_p^{\pm} = -2\phi_x(x,0^{\pm}) \tag{2.9}$$

where C_p^+ and C_p^- correspond to the upper and lower surfaces, respectively.

The TSD equation, given by equation (2.4), is valid for isentropic, irrotational flows about thin shapes (airfoils and wings) immersed in approximately sonic free streams (that is, $M_\infty \sim 1$). For a more detailed discussion of TSD theory, including a discussion of the various three-dimensional formulations, see references 2 and 11.

By using the discriminant test obtained from equation (2.3), it can be shown that equation (2.4) is hyperbolic when

$$\phi_x > \frac{1 - M_\infty^2}{(\gamma + 1)M_\infty^2}$$

and elliptic when

$$\phi_x < \frac{1 - M_\infty^2}{(\gamma + 1)M_\infty^2}$$

In other words, the sign of the first term coefficient determines the equation type. If this coefficient is positive, the local flow is subsonic; if it is negative, the local flow is supersonic. The nonlinearity of the first term is essential for describing the mixed character of transonic flow and is the mechanism by which shock waves are formed.

The characteristic directions associated with the TSD equation are given by

$$\left(\frac{dy}{dx}\right)_{1,2} = \pm[1 - M_\infty^2 - M_\infty^2(\gamma + 1)\phi_x]^{-1/2} \tag{2.10}$$

Notice that these characteristic slopes are symmetrical about the x-axis regardless of the local velocity-vector orientation; that is, the characteristics are not a function of the y-component of the velocity (ϕ_y) (see fig. 4). This situation, which is in dramatic contrast to the full-potential equation, has certain implications regarding spatial-difference approximations for both the TSD and the full-potential equation.

2.3 Full-Potential Equation

The full-potential equation written in conservative form for two-dimensional Cartesian coordinates (x,y) is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y = 0 \tag{2.11a}$$

where the density ρ is defined by

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}(\phi_x^2 + \phi_y^2)\right]^{1/\gamma-1} \tag{2.11b}$$

In equation (2.11), γ is the ratio of specific heats (equal to 1.4 for air) and ϕ is the full or exact velocity potential given by

$$\nabla\phi = \phi_x\hat{i} + \phi_y\hat{j} = u\hat{i} + v\hat{j} = \vec{q} \tag{2.12}$$

where ϕ_x and ϕ_y are the velocity components in the x and y directions, respectively. The density and velocity components appearing in equation (2.11) are nondimensionalized by the stagnation density ρ_s and by the critical speed of sound, a_*. Thus, at a stagnation point,

$$\phi_x, \phi_y \to 0 \ , \qquad \rho \to 1 \tag{2.13}$$

and at a sonic line,

$$\phi_x^2 + \phi_y^2 \to 1 \ , \qquad \rho \to \left[1 - \frac{\gamma - 1}{\gamma + 1}\right]^{1/\gamma-1} = 0.633938145 \ . \ . \ . \tag{2.14}$$

The latter condition is quite useful in providing a simple test for supersonic flow; that is, if $\rho < 0.633938145 \ . \ . \ .$ the flow is supersonic, and if $\rho > 0.633938145 \ . \ . \ .$ the flow is subsonic.

Several useful auxiliary relations are given as follows.

Bernoulli equation (energy equation):

$$\frac{q^2}{2} + \frac{a^2}{\gamma - 1} = \text{constant} \tag{2.15}$$

Isentropic equation of state (perfect gas):

$$P/\rho^\gamma = \text{constant} \tag{2.16}$$

Speed of sound definition:

$$a^2 = \frac{dP}{d\rho} = \frac{\gamma P}{\rho} \tag{2.17}$$

The second equality in equation (2.17) is obtained by using the isentropic equation of state. Using the same nondimensionalization for these realtions as for the full-potential equation yields

$$\frac{q^2}{2} + \frac{a^2}{\gamma - 1} = \frac{1}{2}\frac{\gamma + 1}{\gamma - 1} \tag{2.18}$$

$$\frac{P}{\rho^\gamma} = \frac{\gamma + 1}{2\gamma} \tag{2.19}$$

$$a^2 = \frac{\gamma P}{\rho} \tag{2.20}$$

Note that the density expression shown in equation (2.11b) has been derived from equations (2.18)-(2.20).

The full-potential equation given by equation (2.11) is valid for isentropic, irrotational flows about arbitrary shapes. That is, the full-potential equation is not restricted to thin shapes as is the TSD equation. However, to obtain physically realistic results, the full-potential equation is restricted to shapes and to flows for which viscous effects (separated flow) are not significant. The full-potential equation is also restricted to flows ranging from incompressible ($M_\infty \sim 0$) to transonic ($M_\infty \sim 1$). These potential formulations are valid in transonic flow (even though they are, strictly speaking, isentropic) because weak, or transonic, shock waves can be reasonably approximated by an isentropic formulation. A comparison of the isentropic shock-jump relation with the exact Euler shock-jump relations

(Rankine-Hugoniot relations) is given in figure 5 (see ref. 12). Note that for a local Mach number (M_1) at or below 1.3 a reasonable approximation is maintained by the isentropic assumption.

It is essential that the finite-difference approximation to the full-potential equation be cast in conservative form (see refs. 13 and 14). Otherwise, the shock-capturing procedure will not necessarily conserve mass across the shock wave. The effective mass source added at the shock wave is dependent on nonphysical considerations such as the local grid spacing. Nonconservative schemes have been used, with good success, in transonic-flow simulations for many engineering applications. This is because the effective mass production at shocks fortuitously models shock/boundary-layer interactions. A superior approach is to use conservative form with viscous corrections added via the boundary-layer equations. Therefore, the proper shock strength and position (within the framework of the isentropic formulation), automatically corrected by the proper viscous effects, will be obtained.

Classification of the full-potential equation [eq. (2.11)] is difficult because it is not in standard form [eq. (2.1)]. Transformation of equation (2.11) into nonconservative form will facilitate the classification process. The nonconservative full-potential equation written in two-dimensional Cartesian coordinates is given by

$$(a^2 - \phi_x^2)\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + (a^2 - \phi_y^2)\phi_{yy} = 0 \tag{2.21}$$

where a is the local speed of sound computed from the Bernoulli equation [see eq. (2.18)]. The discriminant [see eq. (2.3)] for equation (2.21) becomes

$$B^2 - 4AC = 4a^2(q^2 - a^2) \tag{2.22}$$

Therefore, the full-potential equation, either in the conservative form [eq. (2.11)] or nonconservative form [eq. (2.21)], is hyperbolic for supersonic flow ($q > a$), parabolic for sonic flow ($q = a$), and elliptic for subsonic flow ($q < a$).

The characteristic directions associated with the full-potential equation are given by

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{-\phi_x\phi_y \pm \sqrt{a^2(q^2 - a^2)}}{a^2 - \phi_x^2} \tag{2.23}$$

Notice that the characteristic directions for the full-potential equation are not symmetric about the x-axis as was the case with the TSD equation. Instead, the characteristics are symmetric about the stream direction. (This fact will become obvious in a subsequent section.) Therefore, like the local velocity vector $\vec{q}$, the characteristic directions can have virtually any orientation, which makes numerical solution of the full-potential equation more complicated than the previously discussed TSD equation. After the presentation of several classical relaxation schemes for purely elliptic equations, we shall return to this and other points associated with the numerical solution of the full-potential equation for transonic flows.

3. CLASSICAL RELAXATION ALGORITHMS

A look at some of the classical relaxation schemes applied to elliptic equations, including point and block (line) iterative techniques, is considered next. Additional details on this subject can be found in Ames (ref. 9). To facilitate this discussion, notation conventions, which will apply throughout these notes, are now presented.

3.1 Notation Conventions

In general, the notation, $\phi_{i,j}^{n}$, will be used to represent the nth iterate of the discretized dependent variable, ϕ, at a position in the finite-difference mesh given by $x = i\Delta x$ and $y = j\Delta y$. When transformed coordinates (ξ,η) are involved, the i and j subscripts will be used to represent position in the computational-domain, finite-difference mesh (that is, $\xi = i$ and $\eta = j$ where $\Delta\xi = \Delta\eta = 1.0$). Definitions for the various difference operators used in these notes are given by the following.

First central difference (second-order accurate):

$$\delta_x(\ )_{i,j} \equiv \frac{1}{2\Delta x}\left[(\ )_{i+1,j} - (\ )_{i-1,j}\right] \tag{3.1}$$

First backward difference (first-order accurate):

$$\overleftarrow{\delta}_x(\ )_{i,j} \equiv \frac{1}{\Delta x}\left[(\ )_{i,j} - (\ )_{i-1,j}\right] \tag{3.2}$$

First forward difference (first-order accurate):

$$\overrightarrow{\delta}_x(\ )_{i,j} \equiv \frac{1}{\Delta x}\left[(\ )_{i+1,j} - (\ )_{i,j}\right] \tag{3.3}$$

Second central difference (second-order accurate):

$$\delta_{xx}(\ )_{i,j} \equiv \frac{1}{\Delta x^2}\left[(\ )_{i+1,j} - 2(\ )_{i,j} + (\ )_{i-1,j}\right] \tag{3.4}$$

Forward shift operator:

$$E_x^{+1}(\ )_{i,j} \equiv (\ )_{i+1,j} \tag{3.5}$$

Backward shift operator:

$$E_x^{-1}(\ )_{i,j} \equiv (\ )_{i-1,j} \tag{3.6}$$

Central average operator:

$$\mu_x(\ )_{i,j} \equiv \frac{1}{2}\left[(\ )_{i+1,j} + (\ )_{i-1,j}\right] \tag{3.7}$$

Forward average operator:

$$\overrightarrow{\mu}_x(\ )_{i,j} \equiv \frac{1}{2}\left[(\ )_{i+1,j} + (\ )_{i,j}\right] \tag{3.8}$$

Backward average operator:

$$\overleftarrow{\mu}_x(\ )_{i,j} \equiv \frac{1}{2}[(\ )_{i,j} + (\ )_{i-1,j}] \tag{3.9}$$

These operator definitions have all been defined using the x-direction. Of course, additional operators can be defined for any space or time coordinate similarly to those for the x-direction. Certain identities associated with these operators exist and can be useful. For example, all difference and averaging operators can be expressed, using shift-operator notation:

$$\begin{aligned} \overleftarrow{\delta}_x &= \frac{1}{\Delta x}(1 - E_x^{-1}) \\ \overleftarrow{\mu}_x &= \frac{1}{2}(1 + E_x^{-1}) \\ \overleftarrow{\delta}_x\overleftarrow{\delta}_x &= \frac{1}{\Delta x}(1 - E_x^{-1})\frac{1}{\Delta x}(1 - E_x^{-1}) = E_x^{-1}\delta_{xx} \\ &= \frac{1}{\Delta x^2}(1 - 2E_x^{-1} + E_x^{-2}) \end{aligned} \tag{3.10}$$

Notice that the symmetrical combination of first-order-accurate, one-sided difference operators creates second-order-accurate, centered-difference operators.

$$\begin{aligned} \delta_{xx} &= \frac{1}{\Delta x^2}(E_x^{+1} - 2 + E_x^{-1}) = \frac{E_x^{+1} - 1}{\Delta x}\,\frac{1 - E_x^{-1}}{\Delta x} \\ &= \overrightarrow{\delta}_x\overleftarrow{\delta}_x = \overleftarrow{\delta}_x\overrightarrow{\delta}_x \\ \delta_x &= \frac{1}{2\Delta x}(E_x^{+1} - E_x^{-1}) = \frac{1}{2}(\overrightarrow{\delta}_x + \overleftarrow{\delta}_x) \end{aligned} \tag{3.11}$$

This fact will be useful in the application of fully implicit, approximate-factorization schemes for solving the full-potential equation.

The classical relaxation schemes discussed next will all be presented for the two-dimensional Laplace equation:

$$\phi_{xx} + \phi_{yy} = 0 \tag{3.12}$$

Each scheme will be put into a standard, two-level correction form (sometimes called delta form) given by

$$NC_{i,j}^n + \omega L\phi_{i,j}^n = 0 \tag{3.13}$$

where $C_{i,j}^n$ is the nth-iteration correction defined by

$$C_{i,j}^n \equiv \phi_{i,j}^{n+1} - \phi_{i,j}^n \tag{3.14}$$

$L\phi_{i,j}^n$ is the nth-iteration residual, which is a measure of how well the finite-difference equation is satisfied by the nth-level solution, ϕ^n; and ω is a relaxation parameter. The general iteration scheme given by equation (3.13) can be considered as an iteration in time (that is, pseudo-time — the scheme does not actually apply to a physical time-dependent differential equation). This consideration allows the n superscript to be regarded as a time-index, that is, $C^n \sim \Delta t \phi_t$. The N-operator determines the type of iteration scheme and, therefore, is the only quantity from equation (3.13) to change from scheme to scheme.

3.2 The von Neumann Test for Stability

The general iteration scheme represented by equation (3.13) can be investigated for numerical stability by using the Fourier, or von Neumann, test for numerical stability. This scheme was developed by von Neumann in the early 1940s and was first discussed in detail by O'Brien et al. (ref. 15) in 1951. Additional details can be found in Smith (ref. 16), Ames (ref. 9), Mitchell (ref. 10), and Richtmyer and Morton (ref. 17).

The propagation of numerical error is studied by substituting a suitable solution,

$$\phi^n_{i,j} = e^{\lambda t} e^{iax} e^{iby} \tag{3.15}$$

into the finite-difference scheme [eq. (3.13)]. In equation (3.15) a and b are wave numbers associated with the Fourier series which represents the solution at $t = 0$, λ is (in general) a complex constant, and the superscript i is $\sqrt{-1}$. The resulting expression is solved for

$$G = e^{\lambda(t+\Delta t)}/e^{\lambda t} = e^{\lambda \Delta t} \tag{3.16}$$

This quantity is called the amplification factor and provides an indication of error growth or decay. If $G > 1$ for any values of a and b, the scheme is numerically unstable. If $G \leq 1$ for all values of a and b, the scheme is stable.

There are several important limitations associated with the von Neumann stability test. First, this test applies only to linear difference schemes with constant coefficients; however, if the difference scheme in question has variable coefficients, the method can still be applied locally with a good chance for accurately predicting numerical stability characteristics. Second, the effect of boundary conditions is neglected by this method (that is, the boundary conditions are assumed to be periodic). Another type of stability analysis, referred to as the matrix method, is preferable when the effect of boundary conditions needs to be analyzed (see refs. 9, 10, 16 for discussion on this topic).

3.3 The Point-Jacobi Scheme

The point-Jacobi relaxation scheme can be expressed as follows:

$$\frac{1}{\Delta x^2}\left(\phi^n_{i+1,j} - 2\phi^{n+1}_{i,j} + \phi^n_{i-1,j}\right) + \frac{1}{\Delta y^2}\left(\phi^n_{i,j+1} - 2\phi^{n+1}_{i,j} + \phi^n_{i,j-1}\right) = 0 \tag{3.17}$$

Rewriting this scheme in standard form yields

$$\left(-\frac{2}{\Delta x^2} - \frac{2}{\Delta y^2}\right)C^n_{i,j} + (\delta_{xx} + \delta_{yy})\phi^n_{i,j} = 0 \tag{3.18}$$

where

$$N = N_{PJ} = -\frac{2}{\Delta x^2} - \frac{2}{\Delta y^2} \tag{3.19}$$

and

$$L = \delta_{xx} + \delta_{yy} \tag{3.20}$$

Note that the N-operator for the point-Jacobi scheme is just a scalar or in matrix form a simple diagonal matrix. Notice also that $\omega = 1$; that is, the point-Jacobi scheme is not overrelaxated. Because of the simple foim of N_{PJ}, obtaining the (n + 1)st-level solution from the nth-level solution is quite easily accomplished. For instance, rewriting equation (3.17) with only the $n + 1$ term on the left and with $\Delta x = \Delta y$ yields

$$\phi^{n+1}_{i,j} = \frac{1}{4}\left(\phi^{n}_{i+1,j} + \phi^{n}_{i-1,j} + \phi^{n}_{i,j+1} + \phi^{n}_{i,j-1}\right) \tag{3.21}$$

Thus, the value at each grid point is simply replaced by the average of the values at the four nearest neighboring points. The simplistic nature of the point-Jacobi scheme and, in particular, the large difference between the construction of the N and L operators, yields very slow convergence, especially as the mesh is refined.

Stability for the point-Jacobi scheme can be investigated by using the von Neumann stability analysis (see sec. 3.2). The amplification factor (for $\Delta x = \Delta y$) is given by

$$G_{PJ} = 1 + \frac{1}{2}(\cos a\Delta x - 1) + \frac{1}{2}(\cos b\Delta y - 1) \tag{3.22}$$

For stability, the magnitude of the amplification factor must be bounded by 1. Notice that G_{PJ} is purely real. This makes the stability analysis somewhat easier. Since both $\cos a\Delta x - 1$ and $\cos b\Delta y - 1$ range from a maximum of zero to a minimum of -2, the growth factor can never be larger than 1 nor smaller than -1. Notice that if ω had been retained, any value above 1 (representing overrelaxation) would have caused instability.

If equation (3.18) is rewritten as

$$\phi^{n+1}_{i,j} - \phi^{n}_{i,j} = \Delta\tau(\delta_{xx} + \delta_{yy})\phi^{n}_{i,j} \tag{3.23}$$

where $\Delta\tau = (2/\Delta x^2 + 2/\Delta y^2)^{-1}$, the point-Jacobi relaxation scheme looks very much like a standard heat-equation integration scheme in which $\Delta\tau$ is the time-integration step-size. The stability condition associated with the time-integration of the heat equation using the present "explicit scheme" is given by

$$\Delta t/\Delta x^2 + \Delta t/\Delta y^2 \leq 1/2 \tag{3.24}$$

This condition is automatically satisfied by the point-Jacobi scheme through the definition of $\Delta\tau$.

3.4 The Point-Gauss-Seidel Scheme

The point-Gauss-Seidel relaxation scheme is very similar to point-Jacobi with one important difference: as we sweep through the mesh, the solution at some points has already been updated. Point-Gauss-Seidel uses this "extra" information by using the latest updated solution at every point possible. For instance, assuming we sweep through increasing values of the mesh indices, i and j, we have

$$\frac{1}{\Delta x^2}\left(\phi^n_{i+1,j} - 2\phi^{n+1}_{i,j} + \phi^{n+1}_{i-1,j}\right) + \frac{1}{\Delta y^2}\left(\phi^n_{i,j+1} - 2\phi^{n+1}_{i,j} + \phi^{n+1}_{i,j-1}\right) = 0 \quad (3.25)$$

Note that the $i-1,j$ and $i,j-1$ terms, as well as the usual i,j term, are all evaluated at the new $n + 1$ iteration level. Rewriting equation (3.25) in standard form yields

$$\left[\frac{1}{\Delta x^2}\left(E_x^{-1} - 2\right) + \frac{1}{\Delta y^2}\left(E_y^{-1} - 2\right)\right]C^n_{i,j} + L\phi^n_{i,j} = 0 \qquad (3.26)$$

where

$$N_{PGS} = \frac{E_x^{-1} - 2}{\Delta x^2} + \frac{E_y^{-1} - 2}{\Delta y^2} \qquad (3.27)$$

and again $\omega = 1$. The residual operator L is defined as before by equation (3.20). The N-operator for the point-Gauss-Seidel scheme (N_{PGS}) provides a more complete approximation to the L-operator than N_{PJ}. Because of this, the point-Gauss-Seidel scheme convergence is faster than that of point-Jacobi. As we shall see, however, other schemes with much faster convergence speed exist.

Because point-Gauss-Seidel has several entries per row, located on or to the left of the diagonal, advancing from the nth-solution level to the (n + 1)st-solution level does not involve typical matrix inversions, for example, tridiagonal and pentadiagonal. However, obtaining the (n + 1)st-solution level is basically a recursive process. Such recursion is difficult (or impossible) to program on vector computers in such a way as to fully utilize the improvement in computational speed offered by the vector architecture. In this context, point-Jacobi, which is completely explicit, that is, nonrecursive, may actually be more efficient than point-Gauss-Seidel. Other relaxation schemes which are vectorizable are discussed in section 7.6.

3.5 The Successive Overrelaxation Scheme

The successive overrelaxation (SOR) scheme can be presented by first writing the point-Gauss-Seidel scheme [eq. (3.25)] in the following form:

$$\frac{1}{\Delta x^2}\left(\phi^n_{i+1,j} - 2\phi^{\overline{n+1}}_{i,j} + \phi^{n+1}_{i-1,j}\right) + \frac{1}{\Delta y^2}\left(\phi^n_{i,j+1} - 2\phi^{\overline{n+1}}_{i,j} + \phi^{n+1}_{i,j-1}\right) = 0 \qquad (3.28)$$

In equation (3.28) the values of $n + 1$ with an over bar are provisional values modified by the following standard relaxation formula:

$$\phi^{n+1}_{i,j} = r\phi^{\overline{n+1}}_{i,j} + (1 - r)\phi^n_{i,j} \qquad (3.29)$$

where r is a relaxation factor. For values of $r > 1$ the scheme is said to be overrelaxed, and for values of $r < 1$ the scheme is said to be underrelaxed. Equations (3.28) and (3.29) can be combined and rewritten in standard form as

$$\left[\frac{1}{\Delta x^2}\left(E_x^{-1} - \frac{2}{r}\right) + \frac{1}{\Delta y^2}\left(E_y^{-1} - \frac{2}{r}\right)\right]C_{i,j}^n + L\phi_{i,j}^n = 0 \qquad (3.30)$$

The N-operator for the SOR scheme is still of the same form as N_{PGS} [eq. (3.27)], although the matrix elements have been modified by the presence of r. Use of the overrelaxation factor greatly improves the convergence rate of the SOR scheme relative to the point-Gauss-Seidel scheme, as we shall see in section 3.8.

Stability of the SOR scheme as applied to Laplace's equation can easily be investigated by the von Neumann analysis. The amplification factor for the SOR scheme is given by

$$G_{SOR} = \frac{-\cos a\Delta x - \cos b\Delta y - \frac{4}{r} + 4 - i(\sin a\Delta x + \sin b\Delta y)}{\cos a\Delta x + \cos b\Delta y - \frac{4}{r} - i(\sin a\Delta x + \sin b\Delta y)} \qquad (3.31)$$

Notice that the imaginary parts of both the numerator and denominator are identical. Therefore, an alternative, but no less restrictive, condition for stability is given by

$$-1 \le \frac{-\cos a\Delta x - \cos b\Delta y - \frac{4}{r} + 4}{\cos a\Delta x + \cos b\Delta y - \frac{4}{r}} \le 1 \qquad (3.32)$$

Assume r is such that the denominator is negative. This yields, after simplification, the following two inequalities:

$$2/r \ge 1 \qquad (3.33)$$

$$-\cos a\Delta x - \cos b\Delta y \ge -2 \qquad (3.34)$$

which are satisfied for all values $a\Delta x$ and $b\Delta y$ providing $0 \le r \le 2$. The latter restriction on r forces the denominator to always be negative, which is consistent with out initial assumption. Thus, the SOR iteration scheme is stable by the von Neumann stability test.

An interesting approach for finding an optimum value for the SOR relaxation parameter was presented by Garabédian (ref. 18; see also ref. 9). The approach consisted of viewing the SOR solution procedure as a time-dependent problem, that is, an artificial-time-dependent problem. Major features of this technique can be outlined as follows: First, using Taylor series derive the modified equation, then assume that Δx and Δt are so small that the solution to the difference equation (or modified equation) behaves like the solution to the corresponding differential equation. The time-dependent PDE can then be solved analytically using a separation of variables technique. The problem is to choose a relaxation factor r such that the time-dependent analytical solution converges as rapidly as possible to the desired steady-state solution. A typical result obtained with this approach using a square domain $(0 < x < \pi$ and $0 < y < \pi)$ is given by

$$r_{opt} = \frac{2}{1 + \Delta x} \qquad (3.35)$$

Thus, the optimum relaxation factor approaches a value of 2 as the finite-difference mesh is refined. This analysis was performed on Laplace's equation, and, although strictly speaking it is not valid for nonlinear problems, the same results apply qualitatively.

3.6 Block or Line Iteration Schemes

Thus far only point iterative schemes have been discussed. Other methods which involve the simultaneous evaluation of more points at the $n + 1$ iteration level are now considered. This additional implicitness provides for faster convergence while requiring only minimal increases in computational work. The first three line schemes discussed are line-Jacobi, line-Gauss-Seidel, and successive line-overrelaxation (SLOR). These schemes, written in standard correction form, are given by the following.

Line-Jacobi:

$$\left[-\frac{2}{\Delta x^2} + \delta_{yy}\right]C_{i,j}^n + L\phi_{i,j}^n = 0 \tag{3.36}$$

Line-Gauss-Seidel:

$$\left[\frac{1}{\Delta x^2}\left(E_x^{-1} - 2\right) + \delta_{yy}\right]C_{i,j}^n + L\phi_{i,j}^n = 0 \tag{3.37}$$

SLOR:

$$\left[\frac{1}{\Delta x^2}\left(E_x^{-1} - \frac{2}{r}\right) + \delta_{yy}\right]C_{i,j}^n + L\phi_{i,j}^n = 0 \tag{3.38}$$

In each case one additional point has been added to the N-operator differencing molecule. This allows the entire y-difference operator to be modeled in the N-operator and requires the inversion of a set of tridiagonal matrix equations for each iteration. Equations (3.36)-(3.38) have all been written for vertical-line relaxation, that is, the tridiagonal matrix extends along the y-direction. An equally valid form can be constructed by using horizontal-line relaxation. The optimal form depends on characteristics of the particular application.

The SLOR scheme [eq. (3.38)] is widely used today in industry as a general relaxation scheme for solving both the TSD and the full-potential equations. A specific SLOR algorithm used to solve the nonconservative full-potential equation will be discussed in section 4.

3.7 The Alternating Direction Implicit Scheme

One technique for achieving even faster convergence than that provided by the SLOR scheme is to use a fully implicit scheme; that is, a scheme in which each point communicates with every other point during each interation. This type of scheme can be constructed using the approximate factorization (AF) philosophy. First, write the N-operator in direct form, that is, N = L, where L is the usual residual operator given by equation (3.20). Note that if the problem is nonlinear, then N must be a linearized approximation to the nonlinear L-operator. The next step in the construction of a suitable AF scheme is to factor the N-operator by using an appropriate factorization, as indicated by

$$N \sim N_1 N_2 \tag{3.39}$$

Usually the AF scheme consists of two factors for two-dimensional problems and three factors for three-dimensional problems. An important idea behind the factorization is that each of the factors N_1 and N_2 must involve only simple banded matrix inversions, thus reducing the computational work per iteration. Then, both the errors associated with the factorization and the linearization (for nonlinear problems) are removed from the solution simultaneously and automatically by iteration.

Of course, using the N-operator directly (before factorization) as the final form of the N-operator will yield a direct scheme which converges in just a single iteration (for linear problems). This type of scheme requires a special inversion algorithm (direct solver) which is much more computationally expensive than, for instance, a set of tridiagonal matrix inversions. For nonlinear problems, which must be considered for transonic applications, the direct-solver scheme loses its single-iteration advantage and must be iterated just as any other standard scheme. Therefore, the direct-solver scheme will not be considered further in these notes. For more information see references 19-23.

The first AF scheme presented is a reformulation of the Peaceman-Rachford alternating direction implicit (ADI) scheme and can be expressed by choosing N as follows (see refs. 9 and 10 for more information on ADI schemes):

$$N_{ADI} = -\frac{1}{\alpha}(\alpha - \delta_{xx})(\alpha - \delta_{yy}) \tag{3.40}$$

where α is an acceleration parameter which may be considered as the inverse of an artificial time-step, $1/\Delta t$. More on the optimal choice of the α parameter will be presented subsequently. In equation (3.40) both the x and y directions are treated implicitly. The N-operator has been written as the product of two tridiagonal matrix factors which when multiplied out yield

$$N_{ADI} = -\alpha - \frac{1}{\alpha}\delta_{xx}\delta_{yy} + L \tag{3.41}$$

This expression contains the original L-operator plus two error terms. The ADI scheme of equation (3.40) can be restated in practical terms using two sweeps as follows.

Sweep 1:

$$(\alpha - \delta_{xx})f^n_{i,j} = \alpha\omega L\phi^n_{i,j} \tag{3.42}$$

Sweep 2:

$$(\alpha - \delta_{yy})C^n_{i,j} = f^n_{i,j} \tag{3.43}$$

In equations (3.42) and (3.43) ω is the relaxation parameter associated with the standard form and $f^n_{i,j}$ is an intermediate result stored at each mesh point in the finite-difference mesh. In the first sweep, the f-array is obtained by solving a tridiagonal matrix equation for each y = constant line. The correction array $(C^n_{i,j})$ is then obtained in the second sweep from the f-array by solving a tridiagonal matrix equation for each x = constant line. This construction of N allows each grid point in the entire mesh to be influenced by every other grid point during each iteration. As a result, much faster convergence can be obtained with this type of algorithm. More discussion on the ADI scheme in particular and AF schemes in general will be presented in section 7.

3.8 Convergence Rate Estimation

A summary of the schemes presented in the preceding sections of this chapter is given in figure 6. The computational molecules of both the N and L operators are shown schematically for each scheme. Note that the L-operator consists of the same five grid points in each case whereas the N-operator is different, varying from only the single central point (i,j) for the point-Jacobi scheme to all five points for the ADI scheme.

Convergence-rate estimates for each of these schemes are now presented. In brief, this estimation technique proceeds as follows (see Ames, ref. 9 for a more detailed discussion). It can be shown that

$$\|e^{n+1}\|/\|e^{n}\| \cong \lambda \tag{3.44}$$

where e^n is the error in the nth-level solution defined by

$$e^n = \phi^n - \phi_{exact} \tag{3.45}$$

and λ is the average spectral radius of the iteration scheme (that is, the ratio of the average maximum eigenvalue at $n + 1$ to the average maximum eigenvalue at n). To achieve a given reduction in the initial error, say ρ, we have

$$\rho = \|e^n\|/\|e^o\| \cong \lambda^n \tag{3.46}$$

Thus, $-\log_{10} \rho$ is the number of orders of magnitude by which the error is reduced in n iterations. Rearranging, we have

$$\text{Num} = \log_{10} \lambda/\log_{10} \rho \tag{3.47}$$

Equation (3.47) gives the number of iterations required to reduce the initial error, e^o, by $-\log_{10} \rho$ orders of magnitude. Ames (ref. 9) evaluates an expression similar to equation (3.47) for each of the relaxation schemes just presented. For a one-order magnitude drop in the error, that is, for $-\log_{10} \rho = 1$, an estimate of the number of iterations (Num) can be obtained and is given for each scheme in table 1. The quantity Δ is the mesh spacing. Note that the line versions of each algorithm are about a factor of 2 faster than the corresponding point counterparts (except SLOR which is $\sqrt{2}$ faster than SOR). More important, the SOR and SLOR schemes are much faster than the other point or line schemes. For example, when $\Delta = 0.01$, SOR is 200 times faster than point-Gauss-Seidel. Thus, the importance of overrelaxation in relaxation schemes in emphasized. Furthermore, ADI is much faster than SLOR; for example, when $\Delta = 0.01$, ADI is over 10 times faster than SLOR. These convergence rate estimates are compared graphically in figure 7. For more details regarding the assumptions of this analysis see Ames (ref. 9).

4. ALGORITHMS FOR THE FULL-POTENTIAL EQUATION: EARLY IDEAS

In this section, priliminary concepts associated with the numerical solution of the full-potential equation are discussed. Because the nonconservative form of the full-potential equation is more revealing to linear analysis and also because it was the first form of the full-potential equation solved, we will restrict our attention solely to this equation in this section. The nonconservative full-potential equation was presented earlier (see sec. 2.3) and is restated here for convenience:

$$(a^2 - u^2)\phi_{xx} - 2uv\phi_{xy} + (a^2 - v^2)\phi_{yy} = 0 \tag{4.1}$$

For purposes of simplifying the analysis we can treat the coefficients $a^2 - u^2$, $-2uv$, and $a^2 - v^2$, as local constants. Thus, the equation is effectively linearized.

4.1 Subsonic Difference Scheme

A spatial finite-difference scheme for the nonconservative full-potential equation (4.1), which is valid for subsonic flow only, is given by

$$L\phi_{i,j} = [(a^2 - u^2)\delta_{xx} - 2uv\delta_x\delta_y + (a^2 - v^2)\delta_{yy}]\phi_{i,j} \tag{4.2}$$

This scheme is centrally differenced and second-order accurate. If the point-Jacobi iteration scheme is now applied to equation (4.2) we have

$$\left[-\frac{2(a^2 - u^2)}{\Delta x^2} - \frac{2(a^2 - v^2)}{\Delta y^2}\right] C^n_{i,j} + L\phi^n_{i,j} = 0 \tag{4.3}$$

Stability can be investigated for this scheme by using the von Neumann test. The amplification factor (G) for $\Delta x = \Delta y$ is given by

$$G = 1 + R(\cos a\Delta x - 1) + S(\cos a\Delta x \sin b\Delta y)$$
$$+ T(\cos b\Delta y - 1) \tag{4.4}$$

where

$$R = \frac{a^2 - u^2}{2a^2 - q^2}, \quad S = \frac{uv}{2a^2 - q^2}, \quad T = \frac{a^2 - v^2}{2a^2 - q^2} \tag{4.5}$$

and q is the speed of the fluid $(\sqrt{u^2 + v^2})$. As before with Laplace's equation, the amplification factor is purely real. For stability, the magnitude of the amplification factor must be less than or equal to 1. When the flow is locally supersonic $(q > a)$, this centrally differenced scheme is unstable. An example demonstrating this situation is presented as follows: Assume the flow is aligned with the x-axis; therefore, $u = q$ and $v = 0$. Equation (4.4) reduces to

$$G = 1 + R(\cos a\Delta x - 1) + \frac{a^2}{2a^2 - q^2}(\cos b\Delta y - 1) \tag{4.6}$$

Since $\cos a\Delta x - 1 \le 0$ and $R < 0$ for $q > a$, it is obvious that $G > 1$. That is, this centrally differenced scheme is unstable for supersonic flow.

Proving stability for subsonic flow is a little more difficult. The amplification factor must satisfy $-1 \le G \le 1$. Working with only the first inequality, we have from equation (4.4) the following condition:

$$R(1 - \cos a\Delta x) - S \sin a\Delta x \sin b\Delta y + T(1 - \cos b\Delta y) \le 2 \tag{4.7}$$

The condition for subsonic flow requires the discriminate of equation (2.3) to be less than or equal to zero. This can be restated as

$$\sqrt{RT} \geq |S| \tag{4.8}$$

This expression is used to replace the S coefficient in the cross-product term and makes the original condition for stability more restrictive. The sign of this cross-product term can be either positive or negative. Clearly, if this term is negative then the inequality is always satisfied. Therefore, assuming the sign to be positive yields

$$R(1 - \cos a\Delta x) + \sqrt{RT}|\sin a\Delta x||\sin b\Delta y| + T(1 - \cos b\Delta y) \leq 2 \tag{4.9}$$

Next, add and subtract the terms necessary to combine the cross-product term in a perfect square. This yields

$$R(1 - \cos a\Delta x) + T(1 - \cos b\Delta y) + \frac{R}{2}\sin^2 a\Delta x + \frac{T}{2}\sin^2 b\Delta y$$

$$- \left[\sqrt{\frac{R}{2}}\,|\sin a\Delta x| - \sqrt{\frac{T}{2}}\,|\sin b\Delta y|\right]^2 \leq 2 \tag{4.10}$$

Because the perfect-square term is always negative, it can be neglected, making the original expression more restrictive. After further simplification the inequality becomes

$$-\frac{R}{2}(1 + \cos a\Delta x)^2 - \frac{T}{2}(1 + \cos b\Delta y)^2 \leq 0 \tag{4.11}$$

The inequality is now seen to always be true. Therefore, the first half of the stability analysis is complete. Looking at the second half yields

$$R(\cos a\Delta x - 1) + S\sin a\Delta x \sin b\Delta y + T(\cos b\Delta y - 1) \leq 0 \tag{4.12}$$

Using a similar substitution on this expression to reduce the cross-product term yields, after simplification,

$$-\frac{R}{2}(1 - \cos a\Delta x)^2 - \frac{T}{2}(1 - \cos b\Delta y)^2 \leq 0 \tag{4.13}$$

This inequality is also always true. Thus, the simple central-difference scheme given by equation (4.2) is stable, providing $q < a$, but is unstable when $q > a$. In the example just considered the iteration scheme was point-Jacobi; however, these conclusions can (theoretically) be reached with any iteration scheme.

4.2 Simple Supersonic Difference Scheme

A scheme suitable for supersonic regions of flow was first introduced by Murman and Cole (ref. 24), for the TSD equation. This approach was later advanced by many researchers including Steger and Lomax (ref. 25), Garabedian and Korn (ref. 26), Bailey and Steger (ref. 27), and Ballhaus and Bailey (ref. 28) for a variety of formulations in both two and three dimensions. The basic idea is as follows: First, determine the local flow type at each grid point (either elliptic or hyperbolic), by central differencing the velocity potential. Then, at subsonic points, use the standard central-difference approximation (second-order accurate), and at supersonic points use difference formulas retarded in the upwind direction. This yields an overall differencing scheme in which the computational domain of dependence correctly reflects the physical domain of dependence.

When the flow is essentially aligned with the positive x-coordinate direction the following simple upwind difference scheme is uitable for supersonic regions of flow (see Jameson, ref. 29):

$$L\phi^n_{i,j} = [(a^2 - u^2)E_x^{-1}\delta_{xx} - 2uv\overleftarrow{\delta}_x\delta_y + (a^2 - v^2)\delta_{yy}]\phi^n_{i,j} \tag{4.14}$$

Upwind evaluation of the ϕ_{xx} term, as in equation (4.14), models the original second-order central-difference approximation plus an upwind-differenced artificial viscosity term. This can be shown as

$$\begin{aligned} E_x^{-1}\delta_{xx}\phi &= (\delta_{xx} - \delta_{xx} + E_x^{-1}\delta_{xx})\phi \\ &= \left(\delta_{xx} - \Delta x \frac{1 - E_x^{-1}}{\Delta x} \delta_{xx}\right)\phi \\ &= (\delta_{xx} - \Delta x\overleftarrow{\delta}_x\delta_{xx})\phi \\ &= \phi_{xx} - \Delta x\phi_{xxx} + O(\Delta x^2) \end{aligned} \tag{4.15}$$

where the $-\Delta x\phi_{xxx}$ is the upwind-evaluated artificial viscosity. [Equation (4.15) can also be derived by a standard Taylor series approach.] The spatial difference scheme given by equation (4.14) is only first-order accurate. Second-order accuracy in subsonic regions and first-order accuracy in supersonic regions are typical characteristics of most successful transonic difference schemes based on potential formulations.

An appropriate iteration algorithm for solving the spatial difference scheme expressed by equation (4.14) is given simply by

$$[(a^2 - u^2)E_x^{-1}\delta_{xx} - 2uv\overleftarrow{\delta}_x\delta_y + (a^2 - v^2)\delta_{yy}]C^n_{i,j} + L\phi^n_{i,j} = 0 \tag{4.16}$$

where the N-operator used here is the same as the L-operator defined in equation (4.14). The velocity components used in both the N and L operators of equation (4.16) are defined by

$$\begin{aligned} u^n_{i,j} &= \delta_x\phi^n_{i,j} \\ v^n_{i,j} &= \delta_y\phi^n_{i,j} \end{aligned} \tag{4.17}$$

and a is obtained from these velocity components and equation (2.18). The iteration scheme of equation (4.16) requires the inversion of a set of tridiagonal matrix equations along the y-direction and is most judiciously implemented by sweeping downstream along the x-direction.

Numerical stability of the simple supersonic algorithm just presented can be investigated using the von Neumann stability analysis. Because the N and L operators are identical, the amplification factor is zero, indicating a direct procedure which converges in one iteration. Even though the solution procedure in the supersonic region is very fast, it is not direct. There are two reasons for this. First, the equation being solved is actually nonlinear and must be iterated to convergence. This factor is not measured by the linear stability analysis. Second, the sonic line boundary supplied to this scheme as initial data must be iterated until the proper position is obtained, a process which may take many iterations. Note that if an over-relaxation factor ($\omega > 1$) had been used for this supersonic marching scheme the convergence rate would have been slower, that is, the amplification factor would not have been zero. Setting all relaxation factors to 1 for supersonic regions is a standard procedure for most line relaxation schemes.

The iteration scheme given by equation (4.16) is essentially a hyperbolic marching scheme. Therefore, it is appropriate to check the marching stability of the scheme. This is accomplished by the usual von Neumann stability test in which

$$\phi_{i,j} = e^{ax} e^{iby} \tag{4.18}$$

is substituted into equation (4.16) to yield, after simplification, a quadratic expression in $e^{-a\Delta x}$. Solving this expression for $e^{-a\Delta x}$ yields

$$e^{-a\Delta x} = 1 - i \frac{B}{A} \sin b\Delta y \pm \sqrt{-\frac{B^2}{A^2} \sin^2 b\Delta y - 2 \frac{C}{A} (\cos b\Delta y - 1)} \tag{4.19}$$

The A, B, and C coefficients are given by

$$A = \frac{a^2 - u^2}{\Delta x^2} , \quad B \frac{uv}{\Delta x \ \Delta y} , \quad C = \frac{a^2 - v^2}{\Delta y^2} \tag{4.20}$$

For stability

$$|G| = |e^{a\Delta x}| \leq 1 \tag{4.21}$$

or equivalently

$$|e^{-a\Delta x}| \geq 1 \tag{4.22}$$

Clearly, the condition for stability will be satisfied if the quantity under the radical in equation (4.19) is always negative. The first term is always negative and the second term is negative providing $C/A < 0$. The quantity C/A is negative as long as the x-marching direction remains supersonic, that is, as long as $u > a$. (Keep in mind that this scheme has been formulated for flows essentially aligned with the x-axis where v is small with respect to u.) Thus, the supersonic marching scheme given by equation (4.16) is stable, providing the x-direction remains hyperbolic. When the x marching direction becomes subsonic ($u < a$) even though the flow is still supersonic ($q > a$), a marching instability is predicted.

A schematic example of this situation, using characteristics, is shown in figure 8. Notice that the computational domain of dependence does not include the entire physical domain of dependence, a situation which must lead to trouble. Near sonic lines, when the flow is only slightly misaligned with the x-axis, instability is predicted. Theoretically, this makes the simple supersonic difference scheme given by equation (4.14) impractical.

The marching instability can also be attributed to a sign change of the artificial viscosity term, $\Delta x(u^2 - a^2)\phi_{xxx}$, which should be positive for stable operation. This term, as mentioned previously, is a consequence of the upwind bias on the differencing scheme in supersonic regions and is the mechanism for allowing physically correct compression shocks while disallowing the physically incorrect expansion shocks.

In practice, the simple supersonic difference scheme is suitable for small regions of supersonic flow in which the flow is nearly aligned in the proper direction. Small regions of flow near the sonic line with negative artificial viscosity are not enough to prevent stable operation (ref. 29). However, when the free stream is supersonic or when a swept-wing calculation is involved, a new technique for handling supersonic regions is necessary.

4.3 Rotated Supersonic Difference Scheme

The concept of rotated differencing was first introduced by Jameson (ref. 29) in 1974 and used for solving the nonconservative full-potential equation for transonic flows about wing geometries. This concept utilizes the ideas already discussed, namely, second-order-accurate central differencing for subsonic flow and first-order-accurate upwind differencing for supersonic flow. However, to remove the directional difficulties associated with the previous simple supersonic differencing scheme, the following coordinate-invariant difference scheme is used for supersonic regions. For two-dimensional cases the nonconservative full-potential equation can be transformed into a local stream and stream-normal coordinate system (s - n), which yields

$$(a^2 - q^2)\phi_{ss} + a^2\phi_{nn} = 0 \tag{4.23}$$

where the s and n coordinates are related to the x and y Cartesian coordinates by

$$x = \frac{u}{q} s - \frac{v}{q} n \tag{4.24}$$

$$y = \frac{v}{q} s + \frac{u}{q} n \tag{4.25}$$

and ϕ_{ss} and ϕ_{nn} are expressed in terms of Cartesian-coordinate derivatives by

$$\phi_{ss} = \frac{1}{q^2} (u^2\phi_{xx} + 2uv\phi_{xy} + v^2\phi_{yy}) \tag{4.26}$$

$$\phi_{nn} = \frac{1}{q^2} (v^2\phi_{xx} - 2uv\phi_{xy} + u^2\phi_{yy}) \tag{4.27}$$

[Note that the characteristic directions associated with eq. (4.23) are always symmetric about the stream direction; see eq. (2.3).]

The algorithm proposed by Jameson (ref. 29), which is based on equation (4.23), is now presented. At hyperbolic points (q > a), central-difference formulas are used for all contributions to ϕ_{nn} [eq. (4.27)]. These formulas are given by

$$\phi_{xx} = \frac{1}{\Delta x^2}\left(\phi^{n}_{i+1,j} - \phi^{n+1}_{i,j} - \phi^{n}_{i,j} + \phi^{n+1}_{i-1,j}\right)$$

$$= \frac{1}{4\Delta x\Delta y}\left(\phi^{n}_{i+1,j+1} - \phi^{n}_{i+1,j-1} - \phi^{n+1}_{i-1,j+1} + \phi^{n+1}_{i-1,j-1}\right) \qquad (4.28)$$

$$\phi_{yy} = \frac{1}{\Delta y^2}\left(\phi^{n+1}_{i,j+1} - 2\phi^{n+1}_{i,j} + \phi^{n+1}_{i,j-1}\right)$$

Contributions to ϕ_{ss} are given by $(q > a)$

$$\phi_{xx} = \frac{1}{\Delta x^2}\left(2\phi^{n+1}_{i,j} - \phi^{n}_{i,j} - 2\phi^{n+1}_{i-1,j} + \phi^{n}_{i-2,j}\right)$$

$$\phi_{xy} = \frac{1}{\Delta x\Delta y}\left(2\phi^{n+1}_{i,j} - \phi^{n}_{i,j} - \phi^{n+1}_{i-1,j} - \phi^{n+1}_{i,j-1} + \phi^{n}_{i-1,j-1}\right) \qquad (4.29)$$

$$\phi_{yy} = \frac{1}{\Delta y^2}\left(2\phi^{n+1}_{i,j} - \phi^{n}_{i,j} - 2\phi^{n+1}_{i,j-1} + \phi^{n}_{i,j-2}\right)$$

Equations (4.28) and (4.29) have been written assuming $u > 0$ and $v > 0$. If these signs are different, equations (4.28) and (4.29) are replaced by similar formulas which retard the difference scheme in the proper upwind direction. The n-superscripts have been determined so as to provide favorable temporal damping to the iteration scheme in supersonic regions. More discussion on this topic will be presented shortly.

The iteration scheme obtained by substituting equations (4.28) and (4.29) into equation (4.23) is expressed in standard correction form by

$$\left\{\frac{M^2 - 1}{q^2}\left[\frac{2u^2}{\Delta x}\overleftarrow{\delta}_x + \frac{2uv}{\Delta x\Delta y}\left(2 - E_x^{-1} - E_y^{-1}\right) + \frac{2v^2}{\Delta y}\overleftarrow{\delta}_y\right]\right.$$

$$\left. - \frac{1}{q^2}\left[-\frac{v^2}{\Delta x}\overleftarrow{\delta}_x + \frac{uv}{\Delta x}E_x^{-1}\delta_y + u^2\delta_{yy}\right]\right\}C^{n}_{i,j} + L\phi^{n}_{i,j} = 0 \qquad (4.30)$$

where the terms inside the first set of square brackets arise from the ϕ_{ss} differencing, and the terms inside the second set of square brackets arise from the ϕ_{nn} differencing. The spatial differencing molecules associated with this scheme are shown in figure 9. All four variations corresponding to the velocity vector originating in the four quadrants are shown. The contribution due to the ϕ_{nn} term is shown by + symbols, and the contribution owing to the ϕ_{ss} term is shown by o symbols. Note that the ϕ_{nn} differencing molecule never changes but that the ϕ_{ss} differencing molecule is automatically upwind biased into the quadrant from which the velocity arises. The transition between these differencing variations is smooth because as one difference operator is switched to another the lead coefficient of that term automatically goes through zero.

The rotated difference scheme just presented, completely removes the marching instability problems associated with the simple upwind scheme presented earlier. However, several new limitations or disadvantages are introduced: (1) the rotated differencing scheme no longer mimics a direct marching scheme and, therefore, slows convergence somewhat; (2) the increased size of the computational molecule, as well as the fact that the scheme is first-order accurate in both directions, increases shock smearing; and (3) this scheme must be swept in the flow direction, a minor limitation associated with general curvilinear meshes. Despite these limitations, this scheme pioneered the way for three-dimensional, transonic flow-field calculations using the full-potential equation and is still the basis for a widely used computer code in the aircraft industry — FL022.

4.4 Temporal Damping

The equivalent time-dependent equation associated with the iteration scheme of equation (4.30) can be written as

$$2\alpha\phi_{st} + 2\beta\phi_{nt} + (M^2 - 1)\phi_{ss} - \phi_{nn} = 0 \tag{4.31}$$

where the α and β coefficients are defined by

$$\alpha = (M^2 - 1)\,\frac{\Delta t}{\Delta x}\,\frac{u}{q} + \frac{\Delta t}{\Delta y}\,\frac{v}{q} \tag{4.32}$$

$$\beta = -\,\frac{1}{2}\,\frac{\Delta t}{\Delta x}\,\frac{v}{q} \tag{4.33}$$

The quantity, ϕ_{st}, built into the iteration scheme of equation (4.30), provides temporal damping. This is very important in supersonic regions and is required to maintain stability. The ϕ_{st} term must be differenced upwind and with the proper sign so as to add to the magnitude of the matrix diagonal. This ensures diagonal dominance for the matrix inversion.

Introducing an independent-variable transformation

$$\begin{aligned} t &= T + \alpha X/(M^2 - 1) - \beta Y \\ s &= X \\ n &= Y \end{aligned} \tag{4.34}$$

for equation (4.31) yields

$$\left(\beta^2 - \frac{\alpha^2}{M^2 - 1}\right)\phi_{TT} + (M^2 - 1)\phi_{XX} - \phi_{YY} = 0 \tag{4.35}$$

For supersonic flows, it can be seen that the coordinate T is space-like (elliptic) and that either X (which is the streamwise coordinate) or Y (which is the stream-normal coordinate) is time-like (hyperbolic). In the physical steady-state problem, X is time-like; therefore, to maintain this situation computationally, the following condition is required

$$\alpha^2 > \beta^2(M^2 - 1) \tag{4.36}$$

Rewriting equation (4.36) using the definitions of α and β [eqs. (4.32) and (4.33)] yields

$$(M^2 - 1)\left(\frac{\Delta t}{\Delta x}\,\frac{u}{q} + \frac{\Delta t}{\Delta y}\,\frac{v}{q}\right)^2 > \frac{1}{4}\left(\frac{\Delta t}{\Delta x}\,\frac{v}{q}\right)^2 \tag{4.37}$$

Generally speaking, the iteration scheme of equation (4.30) has been designed to satisfy the condition of equation (4.37). However, problems arise near sonic lines (as $M^2 - 1 \to 0$). Also, in many iteration schemes the ϕ_{st} temporal damping term is not automatically added. For both of these cases ϕ_{st} can be added explicitly. A suitable term for the present case is given by

$$\varepsilon\left(\frac{\Delta tu}{\Delta xq}\,\phi_{xt} + \frac{v}{q}\,\phi_{yt}\right) \tag{4.38}$$

where the ϕ_{xt} and ϕ_{yt} terms are differenced as follows ($u > 0$, $v > 0$):

$$\phi_{xt} \cong \frac{1}{\Delta t\Delta x}\left(\phi^{n+1}_{i,j} - \phi^{n}_{i,j} + \phi^{n+1}_{i-1,j} - \phi^{n}_{i-1,j}\right) \tag{4.39}$$

$$\phi_{yt} \cong \frac{1}{\Delta t\Delta y}\left(\phi^{n+1}_{i,j} - \phi^{n}_{i,j} + \phi^{n+1}_{i,j-1} - \phi^{n}_{i,j-1}\right) \tag{4.40}$$

The parameter ε in expression (4.38) is a user-specified constant, set large enough for stability but not so large as to excessively slow convergence.

5. TRANSFORMATION AND GRID-GENERATION TECHNIQUES

5.1 Governing Equation Transformation

So far, all examples presented have dealt with only Cartesian coordinates. Usually, before solution algorithms can be implemented, the governing equation must be transformed from Cartesian coordinates into some suitable computational domain. Even applications that use Cartesian coordinates in the computational domain typically require the use of stretching or shearing transformations or both. The primary reason for applying an independent variable transformation to the governing equation is to transform any geometrical surfaces in the problem into constant coordinate lines in the computational domain. Thus, boundary-comdition implementation and grid clustering at geometrical surfaces can be achieved without undue difficulty.

A general, independent variable transformation typically used in conjunction with numerical grid-generation procedures, is given by

$$\begin{aligned} \xi &= \xi(x,y) \\ \eta &= \eta(x,y) \end{aligned} \tag{5.1}$$

where x and y represent the Cartesian-coordinate physical domain and ξ and η the computational domain (see fig. 10). The conservative full-potential equation written in Cartesian coordinates was presented in section 2 and is restated here for convenience as follows:

$$(\rho\phi_x)_x + (\rho\phi_y)_y = 0 \tag{5.2a}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\left(\phi_x^2 + \phi_x^2\right)\right]^{1/\gamma-1} \tag{5.2b}$$

Equation (5.2) transformed by equation (5.1) yields the full-potential equation written in the (ξ,η) computational domain,

$$\left(\frac{\rho U}{J}\right)_\xi + \left(\frac{\rho V}{J}\right)_\eta = 0 \tag{5.3a}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\left(U\phi_\xi + V\phi_\eta\right)\right]^{1/\gamma-1} \tag{5.3b}$$

where

$$U = A_1\phi_\xi + A_2\phi_\eta \tag{5.4a}$$

$$V = A_2\phi_\xi + A_3\phi_\eta \tag{5.4b}$$

$$A_1 = \nabla\xi \cdot \nabla\xi = \xi_x^2 + \xi_y^2 \tag{5.5a}$$

$$A_2 = \nabla\xi \cdot \nabla\eta = \xi_x\eta_x + \xi_y\eta_y \tag{5.5b}$$

$$A_3 = \nabla\eta \cdot \nabla\eta = \eta_x^2 + \eta_y^2 \tag{5.5c}$$

and

$$J = \xi_x\eta_y - \xi_y\eta_x \tag{5.6}$$

The quantities U and V are contravariant velocity components along the ξ and η directions, respectively; A_1, A_2, and A_3 are metric quantities; and J is the Jacobian of the transformation. To evaluate the metric quantities of equations (5.5) and (5.6), the following metric identities are required:

$$J = 1/(x_\xi y_\eta - x_\eta y_\xi) \tag{5.7}$$

$$\left.\begin{aligned} \xi_x &= Jy_\eta , & \eta_x &= -Jy_\xi \\ \xi_y &= -Jx_\eta , & \eta_y &= Jx_\xi \end{aligned}\right\} \tag{5.8}$$

This transformation maintains the strong conservation-law form of the original equation and hence possesses characteristics suitable for a shock-capturing scheme. Additional information about such transformations can be found in references 30-34.

The transformation metric quantities just presented — A_1, A_2, A_3, and J — all have interesting physical interpretations. First of all, the Jacobian, J, can be shown to approximate the inverse of the cell area (in three dimensions the Jacobian approximates the inverse of the cell volume). Thus, a simple test on the sign of the numerically computed Jacobian will reveal if the mesh crosses over on itself (an occasional symptom of some numerically generated meshes which is devastating to the stability of the iteration scheme).

In general, the $(x,y) \rightarrow (\xi,\eta)$ transformation produces a nonorthogonal mesh in the physical domain. A direct measure of the amount of mesh skewness or nonorthogonality is the A_2 metric. Since $A_2 = \nabla\xi \cdot \nabla\eta$, the special case of an orthogonal mesh is realized when all the A_2 metrics are zero. The A_1 and A_3 metrics provide a measure of cell aspect ratio. The A_1 metric is approximately the ratio of arc length along the η-direction to the arc length along the ξ-direction. The A_3 metric is approximately the inverse of A_1. Thus, intimate details regarding a finite-difference mesh can be obtained automatically, without even plotting the grid, by just monitoring the A_1, A_2, A_3, and J quantities.

The three-dimensional form of the full-potential equation written in general curvilinear coordinates (ξ,η,ζ) is given by

$$\left(\frac{\rho U}{J}\right)_\xi + \left(\frac{\rho V}{J}\right)_\eta + \left(\frac{\rho W}{J}\right)_\zeta = 0 \tag{5.9a}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\,(U\phi_\xi + V\phi_\eta + W\phi_\zeta)\right]^{1/\gamma-1} \tag{5.9b}$$

The contravariant velocity components U, V, and W are defined by

$$(U,V,W)^T = H^T H(\phi_\xi,\phi_\eta,\phi_\zeta)^T \tag{5.10}$$

the quantity H is defined by

$$H = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} \tag{5.11}$$

and J = det(H). In equation (5.11), x, y, and z are the usual Cartesian coordinates along the flow, span, and vertical directions, respectively. The ξ, η, and ζ computational coordinates represent wraparound, spanwise, and radial-like directions, respectively.

5.2 The Elliptic Grid-Generation Procedure

Perhaps the most popular new technique for grid generation is the numerical approach. The main idea associated with this technique is to establish a set of curvilinear coordinates by requiring that they be solutions to appropriately formulated PDEs. The properties of the PDEs are such that smooth and regular finite-difference grids result. Coordinate lines of the same family do not cross, and coordinate lines of opposite families intersect orthogonally or nearly orthogonally. Dirichlet boundary conditions are specified such that the body or other desired boundaries are automatically mapped to constant coordinate lines in the physical domain. This guarantees a one-to-one mapping in which the mesh is well-ordered and body-fitted. Mechanisms for achieving general mesh control exist through boundary-condition specification and through the control of various arbitrary coefficients depending on exactly which formulation is used. The numerical mapping procedure is generally valid for both two- and three-dimensional flows.

Many numerical grid-generation techniques have been developed for a wide variety of applications. However, only three schemes will be discussed here. For an extensive survey of this subject see Thompson, et al. (ref. 35). The first scheme has its roots in a number of publications, but has been recently developed and popularized by Thompson et al. (refs. 36-38). This scheme is perhaps the most widely used numerical grid-generation scheme. Poisson's equation is used to define the transformation and is given by

$$\left.\begin{aligned} \xi_{xx} + \xi_{yy} &= P(\xi,\eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi,\eta) \end{aligned}\right\} \tag{5.12}$$

The P and Q right-hand-side quantities are defined as a sum of exponential terms each with several free coefficients. These coefficients can be adjusted by the user to provide different types of mesh size and skewness control. Equations (5.12) are transformed to (and numerically solved in) the computational domain, that is, the roles of (ξ,η) and (x,y) are interchanged such that equations (5.12) become

$$\left.\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= -J^2(Px_\xi + Qx_\eta) \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= -J^2(Py_\xi + Qy_\eta) \end{aligned}\right\} \tag{5.13}$$

where

$$\alpha = x_\eta^2 + y_\eta^2 , \quad \beta = x_\xi x_\eta + y_\xi y_\eta , \quad \gamma = x_\xi^2 + y_\xi^2 \tag{5.14}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{5.15}$$

The finite-difference grid is generated by numerically solving equations (5.13)-(5.15). First, all derivatives are replaced by standard second-order-accurate finite differences. The spatial increments, $\Delta\xi$ and $\Delta\eta$, can be arbitrarily chosen and are usually set to 1. Once boundary and initial values of x and y are specified, the final interior values can be computed by standard relaxation procedures.

An important aspect associated with numerical elliptic-solver techniques is that they have a high degree of controllability. The vast number of free coefficients contained in the P and Q terms is an indication of the large amount of control available to the user. There is an obvious difficulty associated with this flexibility: How can it, in a general and simple way, be made available to the user? This problem poses a difficulty in two dimensions and becomes seemingly insurmountable in three dimensions.

Several researchers have experimented with different aspects of this problem, including Thomas and Middlecoff (ref. 39), Steger and Sorenson (ref. 40), and Sorenson and Steger (ref. 41). In the latter approach, a simplified form of the P and Q terms was adopted and is given by

$$\begin{aligned} P &= P_o(\xi)e^{-a(\eta-\eta_1)} \\ Q &= Q_o(\xi)e^{-b(\eta-\eta_1)} \end{aligned} \tag{5.16}$$

where η_1 corresponds to the η_{min} inner boundary, P_o and Q_o are sets of constants which vary in the ξ direction on the η_{min} boundary, and a and b are constants which control the rate of decay of the P and Q forcing terms into the mesh interior. Thus, application of the forcing terms is restricted only to η_{min} where the control is primarily desired. Steger and Sorenson recognized that two types of control are desired: (1) control of the cell aspect ratio at boundaries and (2) control of cell skewness at boundaries. These two conditions can be expressed mathematically by

$$s_\eta = \left.\frac{ds}{d\eta}\right|_\xi = (x_\eta^2 + y_\eta^2)^{1/2} = \Delta s \tag{5.17}$$

and

$$\nabla\xi \cdot \nabla\eta = x_\xi x_\eta + y_\xi y_\eta = |\nabla\xi||\nabla\eta|\cos\theta \tag{5.18}$$

where s is the arc length along the η lines which intersect the body and θ is the angle of intersection. The condition for orthogonality is $\theta = \pi/2$. Other angle specifications are possible, thus allowing precise control on the cell skewness at the grid boundary. By utilizing equations (5.16)-(5.18), as well as the Dirichlet boundary conditions in x and y, expressions for P_o and Q_o can be derived. Specification of $\Delta s(\xi)$ and $\theta(\xi)$ is all that is required to completely determine a new grid with the desired cell aspect ratio and cell skewness at the boundary.

An example of this technique used to generate a grid about a highly cambered 12-to-1 ellipse is shown in figures 11 and 12. Figure 11 shows global and trailing edge detail of the Laplacian mesh (that is, a mesh with no control) and figure 12 shows the same views with control ($\theta = \pi/2$ and $\Delta s = 0.005$). Note the poor grid quality for the case with no control, especially in the concave portion of the ellipse. The case with control, however, produces a nearly orthogonal grid at the body with a nearly uniform Δs distribution closely approximating the desired value of 0.005.

5.3 The Hyperbolic Grid-Generation Procedure

The next numerical grid-mapping procedure to be discussed is based on a hyperbolic set of governing equations. This type of grid-generation procedure is less well developed than the elliptic procedures but does have several desirable characteristics. Examples of this type of grid-generation procedure include Starius (ref. 42) and Steger and Chaussee (ref. 43). In the latter work, the PDEs used to numerically define the finite-difference mesh are given by

$$\nabla\xi \cdot \nabla\eta = \xi_x\eta_x + \xi_y\eta_y = \phi \tag{5.19}$$

$$\xi_x\eta_y - \xi_y\eta_x = J \tag{5.20}$$

Orthogonality is achieved by simply setting ϕ to zero, and J represents the Jacobian of the transformation (that is, effectively the grid-cell area). In this formulation, both ϕ and J are user-specified functions. This provides a great deal of controllability which is perhaps more direct and easier to implement than in most elliptic PDE formulations. Direct specification of J produces a well-behaved mesh that does not cross over on itself except in the most severe cases. Because this system of equations is hyperbolic in η, a solution can be obtained simply by marching away from initial data specified on the inner boundary. Because iteration is not required, computation time for this technique is very small.

A disadvantage of this technique lies in the lack of direct control over the position and distribution of grid points on the outer boundary ($\eta = \eta_{max}$). However, for external aerodynamic applications this limitation is not severe. Other more fundamental problems lie in the treatment of surface singularities or extension to three-dimensions. Because of the hyperbolic nature of the grid-transformation equations, any singularities imposed in the $\eta = \eta_{min}$ initial data — for example, a slope discontinuity in the geometry — will propagate in the η-direction and perhaps cause problems in the grid interior. Formulations of a suitable algorithm in three dimensions is more difficult because the orthogonality condition [eq. (5.19)] expands to three equations. Thus, a system of four independent equations is created where only three can be used.

Much work needs to be completed before numerical grid-generation schemes based on hyperbolic PDEs can be used routinely for three-dimensional calculations. However, with the advantages of speed and controllability, this grid-generation procedure is an attractive alternative to the elliptic-solver grid-generation technique.

5.4 The Parabolic Grid-Generation Procedure

The final numerical grid-generation procedure presented in these notes is now discussed. This procedure, developed by Nakamura (refs. 44 and 45), is based on a parabolic set of equations given by

$$x_\eta = Ax_{\xi\xi} - 2Bx_{\xi\eta} + S_x \tag{5.21a}$$

$$y_\eta = Ay_{\xi\xi} - 2By_{\xi\eta} + S_y \tag{5.21b}$$

where A and B are positive constants, ξ and η are the usual computational coordinates along and away from the airfoil surface, x and y are the physical domain Cartesian coordinates, and S_x and S_y are source terms given by

$$S_x\big|_{i,j} = (x_{i,NJ} - x_{i,j})/(NJ - j) \tag{5.22a}$$

$$S_y\big|_{i,j} = (y_{i,NJ} - y_{i,j})/(NJ - j) \tag{5.22b}$$

In equations (5.22), the i and j subscripts represent position in the finite-difference mesh ($\xi = i\Delta\xi$ and $\eta = j\Delta\eta$) and $j = NJ$ is the outer boundary. Thus, the effect of the outer boundary ($x_{i,NJ}$ and $y_{i,NJ}$) is included in the S_x and S_y source terms.

Equations (5.21) are parabolic in η and can be marched away from initial data ($x_{i,1}$ and $y_{i,1}$). These initial coordinates represent the airfoil surface, specified by the user, with any smooth distribution. A standard difference scheme involving centered differences for all ξ derivatives and backward differences for all η derivatives is used to discretize equations (5.21). This produces a set of uncoupled tridiagonal matrix equations, which, when inverted, yield values of x and y at $j = 2$. This process is repeated marching away from the airfoil surface, $j = 1$, to the outer boundary, $j = NJ$. As j approaches NJ the x and y values automatically approach the specified outer boundary values of $x_{i,NJ}$ and $y_{i,NJ}$.

An example of this procedure applied to a highly cambered thin ellipse is shown in figure 13 (taken from ref. 44). A view of the entire geometry is shown in figure 13(a) and a close-up of the trailing edge is shown in figure 13(b). Note the regularity of cell size around the inner boundary and lack of cell skewness over the entire mesh. This grid compares quite favorably with the grid presented in figure 12 which was generated with explicit controls on cell size and skewness by the elliptic-solver approach. However, the parabolic scheme requires only a fraction of the computational work.

The parabolic scheme is also easily extended to three dimensions (see ref. 45). One additional equation is required to define the third coordinate and each equation has additional terms; otherwise, the extension is quite straightforward. Because the resulting difference equations are no longer narrow-banded an ADI factorization is used. Thus, in proceeding from one coordinate surface to the next, several iterations are required. However, the basic parabolic grid-generation scheme is still direct, requiring only one sweep from the inner to the outer boundary and, thus, requires only a fraction of the computer time used by the elliptic-solver scheme.

An example of a three-dimensional grid generated by the parabolic scheme about a wing/fuselage geometry is shown in figure 14 (taken from ref. 45). The wing planform plane, the fuselage surface, and a typical fuselage cross-sectional grid are all highlighted in this figure.

5.5 Geometric and Analytic Mapping Procedures

Algebraic and analytical mapping procedures can be simple — for example, involving only a stretching or shearing of the coordinate system. On the other hand, they can be more complicated — for example, conformal mappings. The stretching and shearing transformations can be useful for simple geometries but generally are not sufficient by themselves for more complex airfoil and wing calculations. Conformal mappings do have the generality required for providing good quality, economical grids for reasonably complex, two-dimensional geometries. Many researchers have used conformal mappings to generate arbitrary, orthogonal (or nearly orthogonal meshes when a sheared conformal mapping procedure is used) for a host of two-dimensional applications. A few examples include those of Sells (ref. 46) for an airfoil; Kacprzynski (ref. 47) for an airfoil between wind-tunnel walls; Ives (ref. 48) for a multielement airfoil; Caughey and Jameson (ref. 49) and Chen and Caughey (ref. 50) for axisymmetric inlets with and without a centerbody; and Ives and Liutermoza (ref. 51) for axial-flow turbomachinery cascade applications.

The theory behind conformal mapping techniques is governed by analytic functions of a single complex variable. This theory is well developed, but fundamentally limited to applications in two space dimensions. Nevertheless, some researchers have found ways to use conformal mapping techniques to assist in generating grids for three-dimensional problems. A few examples are given by Jameson (ref. 19) for wings; Jameson and Caughey (ref. 52) and Caughey and Jameson (refs. 53 and 54) for wind/body combinations/ and Ives and Menor (ref. 55) for three-dimensional inlet and inlet-centerbody configurations. The basic approach utilized for three-dimensional grids is to generate a series of two-dimensional grids using standard conformal mapping procedures. Then these grids are "stacked" together in the third dimension to form the final three-dimensional grid. This approach has worked well for geometries with smooth variation in the third dimension but lacks generality. Conformal mapping procedures will continue to be used successfully in a wide variety of applications; however, the anticipated trend for grid-generation procedures will be away from such techniques and toward the more general (although presently less well-developed) numerical or geometric mapping procedures.

The geometric grid-generation procedure has recently received much attention and promises to develop into a practical approach. These procedures are efficient, requiring very little computer time, and have general capabilities regarding coordinate line control. In addition, this type of procedure seems to be general enough for easy extension to three dimensions. Eiseman has presented geometric grid-generation techniques for several two- and three-dimensional configurations (refs. 56-58). The mathematical aspects of this new geometrical grid-generation procedure are developed by Eiseman in references 59 and 60. An example application using this grid-generation procedure for the numerical computation of transonic airfoil flows is given by Pulliam et al. (ref. 61).

Other geometric grid-generation procedures have been presented by McNally (ref. 62), Graves (ref. 63), and Eriksson (refs. 64 and 65). The latter procedure, which can be viewed as a generalized spline interpolation technique, generates a finite-difference grid in two or three dimensions by interpolating geometric data from the domain boundaries. The geometric data required include boundary coordinates and derivatives. In some cases outer domain boundaries can automatically be determined by the geometric transformation procedure and, therefore, do not necessarily have to be specified. This procedure is computationally efficient and has good coordinate line control properties. Transonic wing computations using the Euler equations have been obtained using this grid mapping procedure for a variety of configurations (ref. 66).

The geometric grid-generation procedure has only recently been introduced and, therefore, its ultimate role is difficult to predict. However, the general properties of this class of grid-mapping procedures (e.g., flexibility, computational efficiency, and controllability) suggest a large increase in its use for the generation of grids about complicated configurations.

5.6 Solutions-Adaptive Grid-Generation Schemes

A solution-adaptive grid (SAG) technique is defined to be a grid-generation procedure in which the flow-field solution influences the generation of the grid. Usually this solution influence is designed to produce mesh-point clustering about flow-field gradients and, therefore, produce reduced levels of truncation error. Several researchers have experimented with different types of SAG algorithms, including Dwyer et al. (ref. 67), who developed a procedure for various time-accurate heat-transfer problems; Glowinski (ref. 68), who experimented with optimal grids for incompressible, inviscid flow, using a finite-element technique; and Pierson and Kutler (ref. 69), who determined optimal grid-point distributions for several model problems. Additional work has been presented by Rai and Anderson (refs. 70 to 72), in which the positions of the grid points are determined iteratively along with the solution-dependent variables. The method used to adjust the grid-point positions, forces the solution gradients to be averaged or spread evenly over the finite-difference mesh in the computational domain. This has the effect of reducing the average truncation error.

Other examples of SAG approaches applied to transonic airfoil solutions using the full-potential equation are presented by Ushimaru (ref. 73), Holst and Brown (ref. 74), and Nakamura and Holst (ref. 75). In the latter approach, the surface grid distribution is redistributed by using a second-order ordinary differential equation (ODE) in which the dependent variable is a grid-density function. The forcing function for this ODE is a function of the airfoil surface solution gradient which has to be supplied from an independent solution computed on a standard mesh. In regions of large gradient (for example, at shock waves) the surface grid distribution is automatically clustered, and in regions of small gradient the surface grid distribution is coarsened. The interior grid is generated via interpolation using characteristics of the airfoil surface distribution to appropriately cluster the mesh interior.

An example SAG grid taken from reference 75 is shown in figure 15(a). This grid contains 99 × 25 points and was generated about an NACA 0012 airfoil at a free-stream Mach number of 0.75 and an angle of attack of 2°. For convenience, only the inner 14 lines are shown. The surface-pressure coefficient distribution computed with this grid is compared with a surface solution generated on a much finer standard grid (245 × 56) in figure 15(b). Even though the fine standard grid contains 5.5 times more grid points, the shock is slightly steeper for the SAG solution. For this case despite the fact that the SAG procedure utilized two complete solutions, one on the standard grid and the second on the SAG grid, the SAG procedure was 2 to 3 times faster than the fine-grid calculation.

6. SPATIAL DIFFERENCING SCHEMES

6.1 The Finite-Volume Scheme

The finite-volume spatial difference scheme of Jameson and Caughey (refs. 52-54; 76-77) was first used to solve the conservative full-potential equation in 1977. Since then many applications of this scheme have been made in both two and three dimensions. This scheme, written in two dimensions (for convenience) is given by (see ref. 77)

$$\overleftarrow{\mu}_\eta \overleftarrow{\delta}_\xi \left(\frac{\rho U}{J}\right)_{i+1/2,j+1/2} + \overleftarrow{\mu}_\xi \overleftarrow{\delta}_\eta \left(\frac{\rho V}{J}\right)_{i+1/2,j+1/2} = 0 \tag{6.1}$$

where the averaging ($\overleftarrow{\mu}_\xi$ and $\overleftarrow{\mu}_\eta$) and differencing ($\overleftarrow{\delta}_\xi$ and $\overleftarrow{\delta}_\eta$) operators are defined in section 3.1. The derivatives of x, y, and ϕ with respect to ξ and η are required to compute the density (ρ), the contravariant velocity components (U and V), and the Jacobian (J). These computations are all performed at cell centers $(i+1/2,j+1/2)$ by using

$$(\ )_\xi\big|_{i+1/2,j+1/2} \cong \overleftarrow{\mu}_\eta \overleftarrow{\delta}_\xi (\ )_{i+1,j+1} \tag{6.2a}$$

$$(\ )_\eta\big|_{i+1/2,j+1/2} \cong \overleftarrow{\mu}_\xi \overleftarrow{\delta}_\eta (\ )_{i+1,j+1} \tag{6.2b}$$

The resulting spatial differencing scheme is very compact and requires only a single density evaluation per grid point.

However, this scheme has a tendency to produce oscillatory solutions in which the $i + j$ odd points are decoupled from the $i + j$ even points. This situation can be corrected by adding suitable recoupling terms. The resulting scheme becomes

$$\overleftarrow{\mu}_\eta \overleftarrow{\delta}_\xi \left(\frac{\rho U}{J}\right)_{i+1/2,j+1/2} + \overleftarrow{\mu}_\xi \overleftarrow{\delta}_\eta \left(\frac{\rho V}{J}\right)_{i+1/2,j+1/2} - \frac{\varepsilon}{2} \overleftarrow{\delta}_\xi \overleftarrow{\delta}_\eta (A_\xi + A_\eta) \overrightarrow{\delta}_\xi \overrightarrow{\delta}_\eta \phi_{i,j} = 0 \tag{6.3}$$

where

$$\left.\begin{aligned} A_\xi\big|_{i+1/2,j+1/2} &= \frac{\rho}{J} (A_1 - U^2/a^2)_{i+1/2,j+1/2} \\ A_\eta\big|_{i+1/2,j+1/2} &= \frac{\rho}{J} (A_3 - V^2/a^2)_{i+1/2,j+1/2} \end{aligned}\right\} \tag{6.4}$$

A value of one half is generally used for the constant ε. Addition of these terms recouples the odd and even points and represents a suitable spatial differencing scheme for subsonic regions of flow.

The finite-volume scheme is stabilized in supersonic regions by the explicit addition of artificial viscosity terms given by

$$\hat{P}_{i,j} = \sigma\rho/Ja^2(U^2\delta_{\xi\xi} + UV\delta_\xi\delta_\eta)\phi_{i,j} \tag{6.5a}$$

$$\hat{Q}_{i,j} = \sigma\rho/Ja^2(UV\delta_\xi\delta_\eta + V^2\delta_{\eta\eta})\phi_{i,j} \tag{6.5b}$$

where the switching function σ is defined by

$$\sigma = \max[0,\ 1 - (M_c/M)^2] \tag{6.6}$$

The M_c parameter used in equation (6.6) is a critical Mach number, defined in such a way that the spatial differencing scheme uses the subsonic differencing for values of local Mach number below M_c and the supersonic differencing for values of the local Mach number above M_c. In other words, the transition from central to upwind differencing does not necessarily take place at the sonic line. Note that M_c must be less than or equal to 1 for stability.

The final spatial differencing scheme that is valid for both subsonic and supersonic regions of flow with the odd-even error compensating terms included is given by

$$\overleftarrow{\delta}_\xi\left[\overleftarrow{\mu}_\eta\left(\frac{\rho U}{J}\right)_{i+1/2,j+1/2} + P_{i+1/2,j}\right] + \overleftarrow{\delta}_\eta\left[\overleftarrow{\mu}_\xi\left(\frac{\rho V}{J}\right)_{i+1/2,j+1/2} + Q_{i,j+1/2}\right]$$
$$- \frac{\varepsilon}{2}\overleftarrow{\delta}_\xi\overleftarrow{\delta}_\eta(A_\xi + A_\eta)\overrightarrow{\delta}_\xi\overrightarrow{\delta}_\eta\phi_{i,j} = 0 \qquad (6.7)$$

where the P and Q terms are defined as follows:

$$P_{i+1/2,j} = \begin{cases} \hat{P}_{i,j} & \text{if} \quad U \geq 0 \\ -\hat{P}_{i+1,j} & \text{if} \quad U < 0 \end{cases} \qquad (6.8a)$$

$$Q_{i,j+1/2} = \begin{cases} \hat{Q}_{i,j} & \text{if} \quad V \geq 0 \\ -\hat{Q}_{i,j+1} & \text{if} \quad V < 0 \end{cases} \qquad (6.8b)$$

The spatial differencing scheme given by equations (6.5)-(6.8) is centrally differenced and second-order accurate in subsonic regions where $M < M_c$ and is upwind-differenced and first-order accurate in supersonic regions (or in regions where $M > M_c$). The upwind influence is retained for general curvilinear meshes regardless of the orientation of the velocity vector. Therefore, this conservative spatial differencing scheme approximates the rotated differencing scheme first developed by Jameson (ref. 29) for the nonconservative form of the full-potential equation.

Extensions of this scheme to higher orders of accuracy have been investigated by several researchers, including the work reported in references 76 and 78-81. For example, Caughey and Jameson (ref. 76) modified the P and Q terms such that (looking at only the P term)

$$P_{i+1/2,j} = \begin{cases} \hat{P}_{i,j} - (1 - c\delta_\xi\rho)\hat{P}_{i-1,j} & \text{if} \quad U \geq 0 \\ -\hat{P}_{i+1,j} + (1 - c\delta_\xi\rho)\hat{P}_{i+2,j} & \text{if} \quad U < 0 \end{cases} \qquad (6.9)$$

where c is a constant of order unity. When the local solution is smooth, the overall scheme is formally second-order accurate. In regions of high solution gradient (for example, at shock waves) these added terms force the overall accuracy to revert back to first order. This type of hybrid scheme has been found to be useful for maintaining stability in strong shock calculations.

6.2 The Artificial Density Scheme

The artificial density spatial differencing scheme for the full-potential equation has been independently presented in several different forms (refs. 82-84). These forms, although not identical, have certain similarities which can be attributed to the earlier work of Jameson (ref. 85). Jameson's work is characterized by a scheme with an explicitly added artificial viscosity term. This term biases the spatial difference scheme in the upwind direction for supersonic regions of flow but does not affect the centrally differenced scheme in subsonic regions. The three schemes of references 82-84 use this approach with one basic simplification: the upwind bias is accomplished by an upwind evaluation of the density coefficient. All three procedures compute this upwind or artificial density quantity in different ways.

In the procedure of Holst and Ballhaus (ref. 83; see also refs. 34, 86) the finite-difference approximation for the full-potential equation written in general curvilinear coordinates [see eq. (5.3a)] is given by

$$\overleftarrow{\delta}_\xi\left(\frac{\tilde{\rho}U}{J}\right)_{i+1/2,j} + \overleftarrow{\delta}_\eta\left(\frac{\bar{\rho}V}{J}\right)_{i,j+1/2} = 0 \tag{6.10}$$

where the operators $\overleftarrow{\delta}_\xi(\ )$ and $\overleftarrow{\delta}_\eta(\ )$ are first-order-accurate, backward-difference operators in the ξ and η directions, respectively (see sec. 3.1), and the density coefficients $\tilde{\rho}$ and $\bar{\rho}$ are defined by

$$\tilde{\rho}_{i+1/2,j} = [(1 - \nu)\rho]_{i+1/2,j} + \nu_{i+1/2,j}\rho_{i+r+1/2,j} \tag{6.11a}$$

$$\bar{\rho}_{i,j+1/2} = [(1 - \nu)\rho]_{i,j+1/2} + \nu_{i,j+1/2}\rho_{i,j+s+1/2} \tag{6.11b}$$

The r and s subscripts used in equations (6.11) control the upwind direction of the density coefficients and are defined by

$$\left.\begin{aligned} r &= \pm 1 \quad \text{when} \quad U_{i+1/2,j} \lessgtr 0 \\ s &= \pm 1 \quad \text{when} \quad V_{i,j+1/2} \lessgtr 0 \end{aligned}\right\} \tag{6.12}$$

The switching or transition function ν depends on the local Mach number $M_{i,j}$ and the flow direction and is defined by (e.g., looking at only the x-direction)

$$\nu_{i+1/2,j} = \begin{cases} \max[(M_{i,j}^2 - 1)C,0] & \text{for} \quad U_{i+1/2,j} > 0 \\ \max[(M_{i+1,j}^2 - 1)C,0] & \text{for} \quad U_{i+1/2,j} < 0 \end{cases} \tag{6.13}$$

The quantity C is a user-specified constant usually set to a value between 1 and 2.

The density calculation is performed in a straightforward manner by using a discretized version of equation (5.3b). Values of the density are computed and stored at half points (i.e., at $i + 1/2,j$). Values of ϕ_ξ and ϕ_η required for computing the density at $i + 1/2,j$ are given by

$$\left.\begin{aligned} \phi_\xi\big|_{i+1/2,j} &\cong \overrightarrow{\delta}_\xi\phi_{i,j} \\ \phi_\eta\big|_{i+1/2,j} &\cong \overrightarrow{\mu}_\xi\delta_\eta\phi_{i,j} \end{aligned}\right\} \tag{6.14}$$

The contravariant velocity components (U and V) used in equations (6.10) to (6.12) are computed with standard, second-order-accurate, finite-difference formulas. These quantities are computed and used at $i + 1/2,j$ or $i,j + 1/2$. An example differencing formula for U computed at $i + 1/2,j$ is given by

$$U_{i+1/2,j} = A_1\big|_{i+1/2,j}\overrightarrow{\delta}_\xi\phi_{i,j} + A_2\big|_{i+1/2,j}\overrightarrow{\mu}_\xi\delta_\eta\phi_{i,j} \qquad (6.15)$$

The metric quantities A_1, A_2, and A_3 are computed with fourth-order-accurate, finite-difference formulas and are stored at integer points in the finite-difference mesh. Values required at $i + 1/2,j$, $i,j + 1/2$, etc., are obtained by using simple second-order averages. This metric calculation procedure has produced good results on smooth meshes but suffers on meshes that are not smooth. A superior metric differencing scheme which produces good results on even nonsmooth meshes will be discussed later in this section.

With the spatial differencing scheme just outlined, an upwind influence in supersonic regions is achieved without the explicit addition of an artificial viscosity term. Instead, the stabilizing upwind influence is produced by the upwind evaluation of the density in an otherwise centrally differenced scheme. This approach is significant because it simplifies the technique for including an upwind influence into the residual operator. As in the finite-volume scheme presented in section 6.1, the present artificial density scheme closely approximates the effects of a rotated differencing scheme. This aspect greatly contributes to the stability and reliability of the present algorithm for many difficult test cases.

Another variant of the artificial density spatial differencing scheme has been presented by Hafez et al. (ref. 84). In this scheme, which is designated as an artificial compressibility scheme, the density coefficients in both coordinate directions are defined by

$$\tilde{\rho}_{i,j} = \rho_{i,j} - \nu_{i,j}(\rho_s\Delta s)_{i,j} \qquad (6.16)$$

where

$$(\rho_s\Delta s)_{i,j} = \left(\frac{u}{q}\right)_{i,j}\overset{\rightleftarrows}{\delta}_x\rho\Delta x + \left(\frac{v}{q}\right)_{i,j}\overset{\rightleftarrows}{\delta}_y\rho\Delta y \qquad (6.17)$$

The double-arrow notation indicates a first-order, upwind difference, s is the streamwise coordinate direction, and ν is a switching function defined similarly to equation (6.6) or (6.13).

One difficulty associated with the artificial density spatial discretization philosophy is that if either the switching function ν or the density ρ (or both) are not properly computed, the shock capture process will produce large pre-shock oscillations and poor algorithm reliability. Two guidelines offered by South and Jameson (1979, private communication) that help eliminate this unacceptable behavior are (1) the quantity ν should be evaluated at i,j not at $i + 1/2,j$ as in some formulations; and (2) the density values used in equation (6.11) or (6.16) should be computed at $i+1/2,j+1/2$. The last guideline produces a density computation with a minimum-width differencing module in the streamwise direction. With these two guidelines the existence of pre-shock oscillations is greatly reduced.

Examples showing the magnitude of these effects are shown in figures 16-18. Figure 16 shows two pressure coefficient distributions from reference 86 for an NACA 0012 airfoil at a free-stream Mach number of 0.75 and an angle of attack of 2°. The two solutions correspond to different values of C [see eq. (6.13)]. For both of these calculations, ν is incorrectly evaluated at $i + 1/2,j$ instead of i,j, and ρ is computed directly at grid points instead of at the preferred location $(i+1/2,j+1/2)$. As a result a pre-shock oscillation exists even for relatively large values of C. Figure 17 shows the same comparison with ν correctly evaluated at i,j. The pre-shock oscillation has been eliminated, even for relatively small values of C, but in some cases could still exist for values of ν computed from equation (6.6) (when $M_c = 1.0$). Figure 18 shows a pressure coefficient comparison from South et al., (ref. 87) for a nonlifting NACA 0012 airfoil calculation at a free-stream Mach number of 0.85. The two curves correspond to nodal-point (i,j) and mid-cell $(i + 1/2,j + 1/2)$ density calculations. The ν parameter is correctly evaluated at i,j in both cases and defined by equation (6.6) with $M_c = 1.0$. The mid-cell density calculation clearly gives the superior result with no oscillations.

Many researchers have used one of the artificial-density, spatial-differencing schemes mentioned above because of the simple, reliable way in which the supersonic region is stabilized. A few of these applications include those of Farrell and Adamczyk (ref. 88), Akay and Ecer (ref. 89), and Deconinck and Hirsch (ref. 90), for cascade calculations; Shankar (ref. 91) for supersonic marching problems; Eberle (refs. 92 and 93) for a variety of different applications; and Steger and Caradonna (ref. 94) and Goorjian (ref. 95) for unsteady calculations. Results comparing a number of artificial density scheme variations applied in a finite-element context are presented in Habashi and Hafez (ref. 96).

6.3 Spatial Differencing Schemes Based on Flux-Vector Splitting

Spatial discretization schemes based on the flux-vector splitting models of Godunov (ref. 97) and Engquist and Osher (ref. 98) have recently gained in popularity. Their value in capturing shock waves sharply and in providing good stability properties for a variety of different iteration schemes has been reported by Goorjian and Van Buskirk (ref. 99), Goorjian et al. (ref. 100), Boerstoel (ref. 101), and Slooff (ref. 102). A comparison of these schemes is presented in van Leer (ref. 103) as they apply to the one-dimensional Burger's equation,

$$u_t + F_x = 0 \tag{6.18}$$

where u is the flow velocity, and F is the flux, $u^2/2$ (or for the full-potential equation $F = \rho u$). These spatial discretization schemes applied to the steady part of equation (6.18) can be stated as follows (see ref. 102):

$$\frac{1}{\Delta x}(\bar{\bar{F}}^n_{i+1/2} - \bar{\bar{F}}^n_{i-1/2}) = 0 \tag{6.19}$$

where

$$\bar{\bar{F}}^n_{i+1/2} = F^* - \max(\Delta^+_{i-1/2},\Delta^-_{i+1/2}) \quad \text{(Godunov)} \tag{6.20}$$

$$\bar{\bar{F}}^n_{i+1/2} = F^* - \Delta^+_{i-1/2} - \Delta^-_{i+1/2} \quad \text{(Engquist-Osher)} \tag{6.21}$$

In equations (6.20) and (6.21) F* is the sonic value of F and Δ^+ and Δ^- are defined by

$$\Delta^{+}_{i-1/2} = \begin{cases} F^{*} - F^{n}_{i-1/2} & \text{for} \quad u_{i-1/2} > u^{*} \\ 0 & \text{for} \quad u_{i-1/2} < u^{*} \end{cases} \tag{6.22a}$$

$$\Delta^{-}_{i+1/2} = \begin{cases} 0 & \text{for} \quad u_{i+1/2} > u^{*} \\ F^{*} - F^{n}_{i+1/2} & \text{for} \quad u_{i+1/2} < u^{*} \end{cases} \tag{6.22b}$$

Both of these schemes produce standard discretizations in regions away from sonic lines and supersonic-to-subsonic shock waves. At sonic lines and shock waves these schemes differ from standard schemes and are designed to produce smooth solutions through sonic lines and sharp, monotonic shock waves. The only difference between the Godunov scheme [eqs. (6.19), (6.20), (6.22)] and the Engquist-Osher scheme [eqs. (6.19), (6.21), (6.22)] is in the shock-point operator.

6.4 Free-Stream Consistency Conditions

Grid-generated irregularities, such as mapping singularities, rapid stretching, cell skewness, or grid coarseness, manifest themselves in many realistic configurations. Examples of rapid stretching and cell skewness can be found in the grids about multielement airfoils, wing/fuselage configurations, wing/fuselage/pylon/nacelle configurations, or even simple airfoils when the "O" mesh mapping topology is used. In addition, the solution near the outer computational boundary almost always consists of extremely coarse regions of the grid. Ideally, a stable flow-solver algorithm which can handle all of the above mentioned irregularities, yet provide uniform accuracy over the entire mesh, is desired.

The accurate capture of free-stream flow near the outer computational boundary where the mesh is quite coarse can be a difficult problem. In a formulation that is mapped to the computational domain [see eq. (5.3)], it can be shown that if the metric differencing is implemented properly, the truncation error associated with a free-stream distribution of the dependent variable is zero. That is, free stream is admitted as a solution to the finite-difference equations. This type of procedure was addressed by Pulliam and Steger (ref. 104) for the Euler equations but was not used, because of the small improvements in accuracy obtained on smooth meshes. Thomas and Lombard (ref. 105) and Hindman (ref. 106) also worked with geometrically induced errors associated with the metric differencing and found that certain differencing procedures are better than others.

All of the above work was performed on the Euler equation formulation. Chattot et al. (ref. 107) developed a spatial differencing scheme which contained a perfect free-stream capture characteristic for the full-potential equation. However, in this formulation the full-potential equation was not written in strong conservation-law form, that is, the metrics were written outside the main flux differentiation as follows (see ref. 108):

$$A_1(\rho\phi_\xi)_\xi + A_2[(\rho\phi_\xi)_\eta + (\rho\phi_\eta)_\xi] + A_3(\rho\phi_\eta) + G = 0 \tag{6.23}$$

where A_1, A_2, and A_3 are defined by equation (5.5) and G is defined by

$$G = \nabla^2\xi \ \rho\phi_\xi + \nabla^2\eta \ \rho\phi_\eta \tag{6.24}$$

On smooth meshes, where the metric variation is small, this formulation behaves like conservative form. On nonsmooth meshes, such as one generated by a solution-adaptive procedure, the metric variation at (for example) a shock wave, would be large. This could greatly affect the conservation properties of the spatial differencing scheme.

Flores et al. (ref. 109) presented a free-stream-preserving, spatial-differencing scheme for the conservative full-potential equation written in general curvilinear coordinates [see eq. (5.3)]. Unlike the Euler equation scheme presented in reference 104, which produces perfect free-stream capture with a single free-stream consistency condition, the full-potential equation, in general, requires three conditions. The first condition is associated with the density calculation procedure and is developed as follows: The density can be written solely as a function of the fluid speed. Thus, the exact numerical prediction of free-stream density must result in the exact prediction of the free-stream fluid speed. The fluid speed can be written as

$$q^2 = J^2[(y_\eta\phi_\xi - y_\xi\phi_\eta)^2 + (-x_\eta\phi_\xi + x_\xi\phi_\eta)^2] \tag{6.25}$$

This expression reduces precisely to q_∞^2, if the difference operators used for all ξ-differences involving x, y, and ϕ are the same, and if the difference operators used for all η-differences involving x, y, and ϕ are the same. This can easily be verified by substituting difference operators for all derivatives into equation (6.25) and then using the exact free-stream ϕ distribution to simplify.

The second and third free-stream consistency conditions are associated with the flux calculation. Using the fact that the density is exactly a constant in free-stream flow, the full-potential equation can be rearranged to give

$$\left(\frac{\xi_x\phi_x + \xi_y\phi_y}{J}\right)_\xi + \left(\frac{\eta_x\phi_x + \eta_y\phi_y}{J}\right)_\eta = 0 \tag{6.26}$$

where ϕ_x and ϕ_y are given by

$$\phi_x = \xi_x\phi_\xi + \eta_x\phi_\eta = J(y_\eta\phi_\xi - y_\xi\phi_\eta) = u \tag{6.27a}$$

$$\phi_y = \xi_y\phi_\xi + \eta_y\phi_\eta = J(-x_\eta\phi_\xi + x_\xi\phi_\eta) = v \tag{6.27b}$$

If the difference operators for the ξ-differences of x, y, and ϕ and the difference operators for the η-differences of x, y, and ϕ are the same, respectively, then $\phi_x = u_\infty$ and $\phi_y = v_\infty$. This can be verified from equations (6.27) and is the second free-stream consistency condition. Note that this condition is the same as the first condition, providing the density and flux calculations are performed at the same grid locations. However, the density and flux calculations, in an optimal calculation procedure, are not computed in the same locations (see sec. 6.2), and, therefore, these two conditions have to be satisfied independently.

With $\phi_x = u_\infty$ and $\phi_y = v_\infty$, equation (6.26) can be rewritten as

$$u_\infty(y_{\eta\xi} - y_{\xi\eta}) + v_\infty(x_{\eta\xi} - x_{\xi\eta}) = 0 \tag{6.28}$$

In general, equation (6.28) can be rewritten as two separate equations given by

$$y_{\eta\xi} - y_{\xi\eta} = 0 , \qquad x_{\eta\xi} - x_{\xi\eta} = 0 \tag{6.29}$$

Thus, the finite-difference operators used to obtain the metric quantities must commute. This is the third free-stream consistency condition. The last condition is the same condition stated in reference 104 and was required (by itself) to achieve perfect free-stream capture for the Euler equations.

The three free-stream conditions just presented are satisfied by the two-dimensional finite-volume scheme discussed in section 6.1. However, the extension of this finite-volume scheme to three dimensions does not satisfy all three free-stream flow consistency conditions (see ref. 76).

As stated in reference 109, extension of the first two consistency conditions to three dimensions is straightforward. Extension of the third condition to three dimensions is somewhat more difficult. The following three equations must be satisfied:

$$(y_\eta z_\zeta - y_\zeta z_\eta)_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)_\eta + (y_\xi z_\eta - y_\eta z_\xi)_\zeta = 0 \tag{6.30a}$$

$$(x_\zeta z_\eta - x_\eta z_\zeta)_\xi + (x_\xi z_\zeta - x_\zeta z_\xi)_\eta + (x_\eta z_\xi - x_\xi z_\eta)_\zeta = 0 \tag{6.30b}$$

$$(x_\eta y_\zeta - x_\zeta y_\eta)_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)_\eta + (x_\xi y_\eta - x_\eta y_\xi)_\zeta = 0 \tag{6.30c}$$

That is, once all the derivatives of equations (6.30) are replaced by difference operators, these relations must cancel just as in the analytical case (see ref. 110).

Values of A_1, A_2, A_3, and J in the two-dimensional flux calculations, satisfying both the second and third consistency conditions, were obtained with the same set of difference operators for all primitive metrics, x_ξ, x_η, etc. (see sec. 5.1). This is not the case in three dimensions. Separate formulas for the primitive metrics are required to satisfy the latter two conditions. For example, the A_1 metric quantity can be written

$$A_1 = \xi_x\bar{\xi}_x + \xi_y\bar{\xi}_y + \xi_z\bar{\xi}_z \tag{6.31}$$

The quantities without bars must be computed so as to satisfy free-stream consistency condition two, and the barred quantities must satisfy condition three.

The use of this scheme requires more storage (or a moderate increase in execution time if the metrics are recomputed each iteration). Separate values of the metrics are required at four locations, although not all metrics are required at each of these locations. The minimum number of arrays required for a scheme in which the density is computed at $i + 1/2, j + 1/2, k + 1/2$ (assuming no metric recomputation) is 15, which compares to 7 arrays for a standard scheme using the same assumptions. If densities are computed at $i + 1/2, j, k$ the storage for a perfect free-stream capture algorithm is reduced to 13 arrays. With the size of computer memories rapidly increasing, such storage requirements may not be too difficult to obtain.

Two results showing some of the advantages of schemes that capture free stream perfectly are shown in figures 19 and 20 (taken from ref. 109). Results from a transonic airfoil calculation computed with the TAIR computer code (see ref. 111) are presented in figure 19 for the Korn airfoil at its design condition ($M_\infty = 0.75$, $\alpha = 0.115°$; see ref. 112). Three solutions are presented in figure 19 corresponding to (1) a scheme that satisfies none of the consistency conditions but uses fourth-order-accurate metric differencing (scheme 1), (2) a scheme that satisfies only the first consistency condition (scheme 2), and (3) a scheme that satisfies all three consistency conditions (scheme 3). The TAIR code uses a density and metric numerical smoothing at the trailing edge of airfoils. For this study, the density smoothing is disabled for all three schemes, and the metric smoothing is disabled for schemes 2 and 3. The metric smoothing is still active for scheme 1, for it is required for stability.

In figure 19, the shock location is about the same for all three results. However, the shock strength is different in each case. For this calculation a shock-free solution is expected, and, therefore, the strength of the shock produced is a qualitative estimate of the numerical error associated with each scheme. As expected, the third scheme produces the smallest error based on this criterion. Another benefit associated with the third scheme is the smooth trailing-edge solution. Because the density smoothing has been removed at the trailing edge, schemes 1 and 2 produce oscillations in the trailing-edge pressure distribution. However, the scheme 3 result is oscillation-free, even without the trailing-edge smoothing terms. Thus, satisfying free-stream consistency produces a reduction in global error, as well as a reduction in local error around various grid singularities.

Results from a mesh refinement study are presented in figure 20. The lift coefficient is plotted versus the average mesh spacing for the three schemes outlined above, all applied to an NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2°$. During this study, as the mesh was refined, the ratio of grid points on the airfoil surface to the total number of field points was held fixed. All three curves, representing the three different types of metric differencing, approach the same asymptotic limit as they must to be mathematically consistent. On coarse meshes, the error associated with each scheme is quite different: scheme 3 is the most accurate, scheme 1 is next, and scheme 2 is the least accurate. The scheme 3 improvement in accuracy on coarse meshes represents a highly desirable quality in three-dimensional problems. The behavior between schemes 1 and 2 is somewhat unexpected, since scheme 2 satisfies the first free-stream consistency condition and scheme 1 satisfies none of the consistency conditions. Apparently, the fourth-order-accurate metrics associated with scheme 1 produce smaller levels of error in lift (because the present mesh is smooth) than the second-order-accurate metrics of scheme 2.

6.5 Nonisentropic Full-Potential Formulation

An interesting exposition of various types of potential formulations available for approximating the Euler equations is presented by Klopfer and Nixon (ref. 113). Besides the standard mass-energy formulation in which the momentum is not conserved, other formulations that conserve momentum and energy or mass and momentum are discussed. Crocco's theorem is rederived with suitable conservation errors included for each of these formulations. It is shown that the isentropic assumption is not necessary in conjunction with the velocity potential formulation.

An interesting nonisentropic full-potential formulation is derived in reference 113 such that the resultant shock polar is identical to the Euler shock polar. The nonisentropic potential formulation is given by (two-dimensional Cartesian coordinates)

$$(\rho\phi_x)_x + (\rho\phi_y)_y = 0 \qquad (6.32)$$

$$\rho = \left\{K^{-1}\ 1 - \frac{\gamma - 1}{\gamma + 1}(\phi_x^2 + \phi_y^2)]\right]\right\}^{1/\gamma-1} \tag{6.33}$$

where ϕ is the standard velocity potential, ρ is the fluid density, γ is the ratio of specific heats, and K is a function of the entropy (s) defined by

$$K = e^{(s-s_o)/c_v} \tag{6.34}$$

In equation (6.34) s_o is the free-stream value of the entropy and c_v is the specific heat at constant volume.

Assuming that the shock waves are normal to the u-component of velocity, a locally one-dimensional assumption can be made (although as pointed out in ref. 113, this need not be the case). From equation (6.33) and the Euler shock-jump relations, the following relation for K can be derived:

$$K = \frac{u_1^2 - \frac{\gamma - 1}{\gamma + 1}}{1 - \frac{\gamma - 1}{\gamma + 1} u_1^2}\left(\frac{1}{u_1^2}\right)^{\gamma} \tag{6.35}$$

where u_1 is the normal velocity upstream of the shock. Values of K equal to 1 correspond to free-stream flow; values of K larger than 1 correspond to an entropy increase associated with a shock wave.

Implementation of this scheme requires the ability to detect the position of a shock wave, which is easily accomplished for normal shocks. Then the upstream velocity (u_1) must be extracted from the flow field. Next, K is computed from equation (6.35) for each streamline. This can be done iteratively, but in reference 113 a "C" mesh topology is used and it is assumed that each ξ coordinate line is approximately a streamline. The final modification involves a change in the Kutta condition. The far-field circulation is no longer given by the velocity potential jump at the trailing edge. Now it must be computed from the airfoil circulation and the circulation generated around the wake, which is due to the jump in K across the wake.

Computed results taken from reference 113 are shown in figure 21 for an NACA 0012 airfoil at a free-stream Mach number of 0.80 and an angle of attack of 1.25°. The three curves shown correspond to an Euler solution (ref. 33), a full-potential solution (ref. 34), and a solution from the Klopfer-Nixon nonisentropic potential formulation. For this case, the full-potential solution exhibits a strong shock at the airfoil trailing edge. The local shock Mach number is approximately 1.5, far exceeding the full-potential formulation limitation. The nonisentropic formulation, however, essentially produces the Euler solution. Even the lower-surface shock wave, which was not predicted in the isentropic full-potential formulation, is accurately predicted by the nonisentropic formulation. Thus, Euler-like solutions can be produced with the nonisentropic full-potential formulation in just a fraction of the computer time required by the Euler equations.

Nonunique solutions to the full-potential equation were reported by Steinhoff and Jameson (ref. 114). That is, depending on initial conditions, several drastically different flow-field solutions were obtained for the same airfoil coordinates with the same free-stream conditions. These nonunique solutions only occurred for a range of free-stream Mach numbers that produced relatively strong shock waves. Thus, the cause for the nonuniqueness may be associated with the large disagreement that exists between the isentropic full-potential and Euler shock polars for strong shock waves. If this is the case, the nonisentropic full-potential formulation of Klopfer and Nixon, which possesses the Euler shock polar, may represent a solution to the nonunique full-potential problem.

6.6 Other Spatial Differencing Schemes

Many other spatial differencing schemes suitable for the solution of transonic flow problems based on potential or potential-like formulations have been presented. A few of these schemes are briefly discussed in this section. Of particular note are the field panel method of Piers and Slooff (ref. 115); the finite-element method of Vigneron et al. (ref. 116), and the penalty function method described by Bristeau et al. (ref. 117), Periaux (ref. 118), and Bristeau et al. (ref. 119). In the last approach, a least-square finite-element formulation is used to discretize the full-potential equation in conservative form. To exclude expansion shocks the least-square functional is modified to include a penalty function. This penalty function takes on large values for solutions containing nonphysical expansion shocks, that is, for solutions with streamwise positive jumps in velocity, and small values for solutions with proper entropy increasing shocks. In a sense, this penalty function approach is a dissipative device similar to artificial viscosity that is designed to exclude physically unrealistic expansion shocks.

Another approach suitable for the prediction of transonic flow fields is based on the stream function formulation. Early pioneering work was done in this area by Emmons (refs. 120-122). More recently, Hafez and Lovell (ref. 123) presented a stream-function formulation suitable for solving transonic flow. The stream-function equation is given by (two-dimensional Cartesian coordinates)

$$\left(\frac{\psi_x}{\rho}\right)_x + \left(\frac{\psi_y}{\rho}\right)_y = -\omega \qquad (6.36)$$

where

$$\rho = e^{-\Delta s}\left[1 - \frac{\gamma - 1}{2} M_\infty^2 \left(\frac{\psi_x^2 + \psi_y^2}{\rho^2} - 1\right)\right]^{1/\gamma-1} \qquad (6.37)$$

and ω is the vorticity which is computed from the entropy rise across the shock (Δs).

In this approach the stream-function equation is discretized using the artificial density concept of reference 84 to stabilize the scheme in supersonic regions. A unique solution to the doubled-valued density problem, typically associated with transonic applications of the stream-function formulation [see eq. (6.37)], is presented by Hafez and Lovell. Since the irrotationality assumption does not have to be made in the stream-function formulation, results with vorticity and entropy increases across shock waves can be computed. This is highly attractive since solutions to the Euler equations can be simulated with this formulation while only requiring the numerical solution of a single second-order PDE. In addition, because the form of the stream-function equation and the full-potential equation are nearly the same, most solution schemes for the potential formulation can be used to solve the stream-function formulation.

The two drawbacks associated with the stream-function formulation are its increased complication caused by the double-valued density relation and difficulty in extending it to three-dimensional flows. A three-dimensional formulation is presented in reference 123 but not implemented. The formulation expands to three equations which are probably still easier to solve than the Euler equations; however, until comparisons are made, final conclusions cannot be drawn.

7. ITERATION SCHEMES

The next subject of discussion involves the iterative process by which the initial solution is evolved into the final solution. The iteration scheme is primarily responsible for the amount of computational work associated with each algorithm through the number of iterations required for convergence. In recent years, many researchers have experimented with different techniques for reducing the number of iterations associated with transonic flow computations. In this section we will look at the iteration schemes that have proved to be most successful in reducing the computational cost relative to older more standard iterative schemes such as SLOR.

7.1 The Alternating Direction Implicit Scheme

The alternating direction implicit (ADI) scheme has already been discussed (sec. 3.7) for solving Laplace's equation. In this section, an ADI scheme suitable for solving the conservative full-potential equation for transonic flow is presented. The ADI factorization used here is basically the same as that discussed in section 3.7 and can be stated by writing the standard N-operator as follows (see also ref. 108):

$$NC^n_{i,j} = -\frac{1}{\alpha}(\alpha - \overrightarrow{\delta}_\xi A_i \overleftarrow{\delta}_\xi)(\alpha - \overrightarrow{\delta}_\eta A_j \overleftarrow{\delta}_\eta)C^n_{i,j} \qquad (7.1)$$

where α is an acceleration parameter (to be discussed shortly) and A_i and A_j are defined by

$$A_i = \left(\frac{\tilde{\rho}A_1}{J}\right)_{i-1/2,j} \qquad A_j = \left(\frac{\bar{\rho}A_3}{J}\right)_{i,j-1/2} \qquad (7.2)$$

In equation (7.2) the density coefficients, $\tilde{\rho}$ and $\bar{\rho}$ are defined by equation (6.11) and A_1, A_3, and J are metric quantities defined by equations (5.5) and (5.6). The ADI scheme of equation (7.1) is implemented in a two-step format given by the following.

Step 1:

$$(\alpha - \overrightarrow{\delta}_\xi A_i \overleftarrow{\delta}_\xi)f^n_{i,j} = \alpha\omega L\phi^n_{i,j} \qquad (7.3a)$$

Step 2:

$$(\alpha - \overrightarrow{\delta}_\eta A_j \overleftarrow{\delta}_\eta)C^n_{i,j} = f^n_{i,j} \qquad (7.3b)$$

In equations (7.3), ω is a standard relaxation factor and $f^n_{i,j}$ is an intermediate result stored over the entire finite-difference mesh. The residual, $L\phi^n_{i,j}$, is defined by equation (6.10). Step 1 consists of a set of tridiagonal matrix equations along the ξ direction, and step 2 consists of a set of tridiagonal matrix equations along the η direction. The construction of this ADI scheme does not automatically provide the necessary ϕ_{st} to stabilize supersonic regions. However, this type of term can still be included by adding

$$\mp\beta_\xi |U_{i,j}| \overset{\rightleftarrows}{\delta}_\xi \quad \text{and} \quad \mp\beta_\eta |V_{i,j}| \overset{\rightleftarrows}{\delta}_\eta \tag{7.4}$$

inside the parentheses of the first and second steps, respectively. The double-arrow notation on the operators of expressions (7.4) indicate that the difference direction is always upwind, and the sign is chosen so as to increase the magnitude of the matrix diagonal coefficient. The contravariant velocity component scaling used in expressions (7.4) provides a smooth transition from forward to backward difference directions when the flow direction changes sign. The β_ξ and β_η coefficients are constants specified by the user according to need.

Stability of the ADI scheme as given by equations (7.1)-(7.3) can be investigated by considering a simplified form given by

$$-(\alpha - \delta_{xx})(\alpha - \delta_{yy})C^n_{i,j} + \alpha\omega(\delta_{xx} + \delta_{yy})\phi^n_{i,j} = 0 \tag{7.5}$$

This is essentially the scheme presented in section 3 for solving Laplace's equation. Other, more complex model equations can be used, but the present one will allow essentially the same conclusions with less work. The amplification factor for the algorithm of equation (7.5) is given by

$$G_{ADI} = \frac{a_1 + a_2 - \omega a_2}{a_1 + a_2} \tag{7.6}$$

where

$$a_1 = \alpha + \frac{4}{\alpha\Delta x^2 \Delta y^2}(\cos a\Delta x - 1)(\cos b\Delta y - 1) \geq 0 \tag{7.7}$$

$$a_2 = -\frac{2}{\Delta x^2}(\cos a\Delta x - 1) - \frac{2}{\Delta y^2}(\cos b\Delta y - 1) \geq 0 \tag{7.8}$$

The purely real amplification factor is always less than or equal to 1, providing $0 \leq \omega \leq 2$ and $\alpha \geq 0$. Because the only condition for stability on the α parameter is that it be positive, the ADI scheme is said to have unconditional linear stability, as expected for an implicit scheme.

The ADI amplification factor can be factored into a special form given by (for $\omega = 2$):

$$G_{ADI} = \frac{\left[\alpha - \frac{2(1 - \cos a\Delta x)}{\Delta x^2}\right]\left[\alpha - \frac{2(1 - \cos b\Delta y)}{\Delta y^2}\right]}{\left[\alpha + \frac{2(1 - \cos a\Delta x)}{\Delta x^2}\right]\left[\alpha + \frac{2(1 - \cos b\Delta y)}{\Delta y^2}\right]} \tag{7.9}$$

Note that the x and y directions decouple. Thus, the amplification factor can be minimized for a particular eigenvalue associated with (for example) the x-direction by choosing α to satisfy

$$\alpha = \frac{2}{\Delta x^2}(1 - \cos a\Delta x) \tag{7.10}$$

The amplification factor for this value of α and for this eigenvalue is actually zero. For a problem with NI grid points in the x-direction, corresponding to NI eigenvalues, the solution will converge to zero error after NI iterations. Of course, this is only true for a linear problem. Precise estimation of the eigenvalues for a nonlinear problem is generally not possible. Instead, a repeating sequence of α's is used with each element of the sequence chosen to maintain small values of $|G|$ for a given range of eigenvalues (ref. 124). A suitable sequence of α's presented in Ballhaus et al. (ref. 125) is given by

$$\alpha_k = \alpha_H(\alpha_L/\alpha_H)^{k-1/M-1} \qquad k = 1, 2, 3, \ldots, M \tag{7.11}$$

where M is the number of elements in the sequence. The sequence endpoints can be estimated by using equation (7.10). For instance, the lowest eigenvalues, corresponding to low-frequency errors, are approximately given by $a \sim 1$, which yields

$$\begin{aligned} \alpha_L &\cong \frac{2}{\Delta x^2}(1 - \cos a\Delta x) \\ &\cong \frac{2}{\Delta x^2}\left[1 - \left(1 - \frac{\Delta x^2}{2} + \frac{\Delta x^4}{24} \mp \ldots\right)\right] \\ &\cong 1 \end{aligned} \tag{7.12}$$

For high-frequency errors $a\Delta x \sim \pi$, which yields

$$\alpha_H \cong \frac{4}{\Delta x^2} \tag{7.13}$$

In practice, it is well advised to "optimize" both α_L and α_H by trial-and-error numerical experimentation. Of course, this has to be done only once for each code, for α_L and α_H do not strongly depend on the characteristics of the solution being computed.

The ADI scheme just presented has been used to compute transonic flow in a number of different applications; however, the results will not be presented until after the next section. The iteration scheme presented in the next section, AF2, will then be compared and contrasted with the ADI scheme in section 7.3.

7.2 The AF2 Approximate Factorization Scheme

The AF2 scheme was first presented in reference 126 for solving the low-frequency (unsteady) TSD equation. This algorithm was subsequently applied to the solution of the steady TSD equation (ref. 125) and the conservative full-potential equation (refs. 83 and 84). The AF2 fully implicit scheme can be expressed by choosing the standard N-operator as follows:

$$NC_{i,j}^n = -\frac{1}{\alpha}(\alpha - \overrightarrow{\delta}_\eta A_j)(\alpha\overleftarrow{\delta}_\eta - \overrightarrow{\delta}_\xi A_i \overleftarrow{\delta}_\xi)C_{i,j}^n \tag{7.14}$$

where, as with the ADI scheme, A_i and A_j are defined by

$$A_i = \left(\frac{\tilde{\rho}A_1}{J}\right)_{i-1/2,j} \qquad \text{and} \qquad A_j = \left(\frac{\bar{\rho}A_3}{J}\right)_{i,j-1/2} \tag{7.15}$$

In equation (7.15) the density coefficients, $\tilde{\rho}$ and $\bar{\rho}$, are defined by equation (6.11) and A_1, A_2, and J are defined by equations (5.5) and (5.6). The AF2 scheme is implemented in a two-step format given by the following.

Step 1:

$$(\alpha - \overrightarrow{\delta}_{\eta} A_j) f^n_{i,j} = \alpha\omega L\phi^n_{i,j} \tag{7.16a}$$

Step 2:

$$(\alpha\overleftarrow{\delta}_{\eta} - \overrightarrow{\delta}_{\xi} A_i \overleftarrow{\delta}_{\xi}) C^n_{i,j} = f^n_{i,j} \tag{7.16b}$$

In equations (7.16), ω is the usual relaxation factor, α is a convergence acceleration parameter cycled over a sequence of values [see eq. (7.11)], and $f^n_{i,j}$ is an intermediate result stored over the entire finite-difference mesh. Step 1 consists of a set of bidiagonal matrix equations along the η-direction, and step 2 consists of a set of tridiagonal matrix equations along the ξ-direction. With the AF2 factorization, the η-difference approximation is split between the two steps. This generates a $\phi_{\eta t}$-type term, which is useful to the iteration scheme as time-like dissipation. The split η term also places a sweep direction restriction on both steps, namely, in the negative η-direction for the first step [eq. (7.16a)] and in the positive η-direction for the second step [eq. (7.16b)]. Flow direction imposes no sweep direction restrictions on either of the two sweeps.

The N-operator, as presented in equation (7.14) (see ref. 34), is somewhat different from the AF2 scheme presented in references 83 and 125. For the AF2 factorization, the N-operator must be written so that either the ξ- or the η-difference operator is split between the two factors. This construction generates either a $\phi_{\xi t}$-type or a $\phi_{\eta t}$-type term and, if properly differenced, provides time-dependent dissipation to the convergence process. For the "O" mesh topology (see fig. 10), an algorithm with the ξ-direction split produces a $\phi_{\xi t}$ term which is properly differenced either above or below the airfoil. Since the supersonic zone can generally exist on both the upper and the lower airfoil surfaces and since the supersonic zone is usually the most difficult in which to maintain computational stability, it is desirable to keep the $\phi_{\xi t}$ term differenced in the upwind direction on both the upper and lower surfaces. Thus, the N-operator presented in equation (7.14) from reference 34 splits the η-direction. This allows control of the $\phi_{\xi t}$ term because it is added explicitly and is not part of the factorization.

Of course, with this N-operator construction, the $\phi_{\eta t}$ term is upwind differenced in the forward half of the mesh and downwind differenced in the aft half. For this formulation, adverse effects for cases with supersonic flow at the trailing edge may be anticipated but none have been experienced. In fact, cases with free-stream Mach numbers near un'ty have been computed, in which the trailing edge is entirely embedded in supersonic flow with no adverse effects (ref. 86). The precise reason for this behavior is unclear. The spatial differencing scheme, which is differenced in the upwind direction, and the explicitly added $\phi_{\xi t}$ term, which is also always differenced in the proper manner, may overshadow any adverse effects introduced from the $\phi_{\eta t}$ term.

The $\phi_{\xi t}$-type term is included (if necessary) by adding

$$\mp\alpha\beta\overset{\rightleftharpoons}{\delta}_{\xi} \tag{7.17}$$

inside the parentheses of step 2 [see eq. (7.16b)]. As with the ADI scheme, the double-arrow notation indicates that the difference direction is always upwind, and the sign is chosen so as to increase the magnitude of the matrix diagonal coefficient. The parameter β is fixed at a value of 0.3 in subsonic regions. In supersonic regions, β is initialized according to user specifications, for example, to 4.5, and then updated using special logic (see ref. 111) given by

$$\left.\begin{array}{llll} \text{If} & \text{RATIO} < 2.0 & \text{then} & \beta^n = 0.98\ \beta^{n-1} \\ \text{If} & \text{RATIO} > 2.1 & \text{then} & \beta^n = 1.1\ \beta^{n-1} \\ \text{If} & \beta^n > \text{BHIGH} & \text{then} & \beta^n = \text{BHIGH} \\ \text{If} & \beta^n < \text{BLOW} & \text{then} & \beta^n = \text{BLOW} \end{array}\right\} \tag{7.18}$$

where

$$\text{RATIO} = \frac{\text{RAVG}^n}{\text{RAVG}^{n-M}} + \frac{\text{RMAX}^n}{\text{RMAX}^{n-M}} \tag{7.19}$$

$$\text{BHIGH} = \beta^1 + 1 \ , \qquad \text{BLOW} = \beta^1 - 1 \tag{7.20}$$

In equation (7.19), M is the number of elements in the α sequence, and RAVG is the nth iteration average residual. The logic defined by equations (7.18)-(7.20) monitors solution convergence through the parameter RATIO. If convergence is progressing satisfactorily, β is reduced; if not, β is increased. The parameters BHIGH and BLOW are upper and lower bounds which limit the amount of β variation. This update scheme for β is similar to the scheme presented by South et al. (ref. 87).

In addition to the above logic, other larger increases (or decreases) in β are possible. During the iteration process, the base value of β, including the values of BHIGH and BLOW, are increased or decreased, if the developing solution requires more or less time-like dissipation. This logic, largely developed on a trial-and-error basis, automatically keys on the growth rates of the number of supersonic points and the amount of circulation. If these quantities grow rapidly, then β, BHIGH, and BLOW are all increased; if they grow slowly, then these quantities are decreased. Thus, with this type of logic, the time-like dissipation can be automatically adapted to each individual solution.

Stability of the AF2 iteration scheme [eqs. (7.14)-(7.16)] can be investigated by considering a simplified algorithm given by

$$(\alpha\overleftarrow{\delta}_x - \delta_{yy})(\alpha - \overrightarrow{\delta}_x)C^n_{i,j} + \alpha\omega(\delta_{xx} + \delta_{yy})\phi^n_{i,j} = 0 \tag{7.21}$$

This is essentially a scheme for solving Laplace's equation. The standard von Neumann test yields an amplification factor given by

$$G_{AF2} = \frac{a_1 + (1 - \omega)a_2}{a_1 + a_2} \tag{7.22}$$

where

$$a_1 = \frac{\alpha}{\Delta x}\,(1 - e^{-ia\Delta x}) + \frac{2}{\alpha\Delta x\Delta y^2}\,(\cos b\Delta y - 1)(e^{ia\Delta x} - 1) \tag{7.23}$$

$$a_2 = -\frac{2}{\Delta x^2}(\cos a\Delta x - 1) - \frac{2}{\Delta y^2}(\cos b\Delta y - 1) \geq 0 \qquad (7.24)$$

Unlike the ADI scheme amplification factor, which was purely real, the AF2 amplification factor is complex and, therefore, a little more difficult to analyze. However, the imaginary parts in both the numerator and denominator are the same. Thus, an equivalent condition of stability is given by

$$-1 \leq \frac{a_3 + (1 - \omega)a_2}{a_3 + a_2} \leq 1 \qquad (7.25)$$

where

$$a_3 = \mathrm{Re}(a_1)$$

$$= \frac{\alpha}{\Delta x}(1 - \cos a\Delta x) + \frac{2}{\alpha\Delta x\Delta y^2}(\cos b\Delta y - 1)(\cos a\Delta x - 1) \geq 0 \qquad (7.26)$$

The expression given by equation (7.25) is of the same form as the amplification factor for the ADI scheme [eq. (7.6)] and, therefore, is always satisfied, providing $0 \leq \omega \leq 2$ and $\alpha \geq 0$. Thus, the AF2 scheme has the unconditional linear stability typically associated with a fully implicit iteration scheme.

There is an interesting aspect of the stability of the present AF2 scheme associated with the airfoil surface boundary condition, as discovered by South (ref. 127). The residual operator airfoil surface boundary condition is that of flow tangency and is implemented by reflection. Another aspect of the surface boundary condition is that the η-direction difference on f at the airfoil surface is arbitrarily set to zero. South noted that a stability analysis using the proper airfoil surface boundary conditions (with reflection condition included) produces the following stability condition:

$$\alpha \geq \rho A_3 \omega \qquad (7.27)$$

That is, the α parameter (or equivalently the inverse of the time-step) is restricted at the airfoil surface by the condition given in equation (7.27). The quantity A_3 is effectively the cell aspect ratio (see sec. 5.1). Thus, as the mesh is clustered toward the airfoil surface, the stability condition (7.27) becomes more restrictive and, unless the α sequence is suitably modified, divergence can result.

South presented a solution to this problem which consisted of an appropriate modification of the N-operator at the airfoil surface. By modeling the surface N-operator after the surface residual operator, unconditional linear stability was restored.

Finding values of α that cause the AF2 amplification factor to be zero for a given eigenvalue is difficult because the AF2 amplification factor is complex. However, values of α can be determined which do minimize the AF2 amplification factor (see ref. 125) and are given for the low- and high-frequency limits as follows:

$$\alpha_L \cong 1\ , \qquad \alpha_H \cong 1/\Delta \qquad (7.28)$$

Notice that the high-frequency endpoint for the AF2 scheme is quite different from that of the ADI scheme but that the low-frequency values are the same. Again, in practice, it is well advised to optimize both endpoints by trial-and-error numerical experimentation. This, however, need only be done once for each code, for solution parameters such as Mach number, angle of attack, and airfoil coordinate variations do not greatly affect the optimal α sequence.

The AF2 algorithm discussed above has been coded into a user-oriented computer code called TAIR (Transonic AIRfoil analysis). Computational results produced with TAIR are now presented. The first result involves the supercritical Korn airfoil at a free-stream Mach number of 0.74 and an angle of attack of 0°. Pressure coefficient distributions for this slightly off-design case are compared in figure 22, with a result from the GRUMFOIL computer code (ref. 128). The GRUMFOIL computer code has available a viscous correction option which was not used for this calculation. The two results are in excellent agreement. The rms error convergence history curves for this calculation are presented in figure 23. The rms error at iteration n (E^n_{RMS}) is defined by

$$E^n_{RMS} = \left[\frac{\sum_{i=1}^{NI} (C^n_{P_i} - \bar{C}_{P_i})^2}{NI}\right]^{1/2} \tag{7.29}$$

where $C^n_{P_i}$ is the surface-pressure coefficient at the ith grid point and the nth iteration; $\bar{C}_{P_i}$ is the surface-pressure coefficient at the ith grid point taken from the converged solution; and NI is the total number of surface grid points. Using E_{RMS} to compare convergence performance is a much more quantitatively correct procedure than using the standard maximum residual quantity. (More discussion of this point can be found in refs. 83 and 129 and in sec. 7.3 of these notes.) The three curves shown in figure 23 correspond to the following iteration schemes: (1) AF2, (2) hybrid, and (3) SLOR. The hybrid scheme is a combination semidirect/SLOR iteration scheme developed by Jameson (ref. 85), which is composed of one semidirect-solver iteration followed by several SLOR iterations. The purpose of the SLOR iterations is to smooth high-frequency errors generated by the direct-solver step in regions of supersonic flow. For this calculation, the AF2 scheme convergence rate is about 5 times faster than that of the hybrid scheme and about 10 times faster than the SLOR rate.

Additional results obtained from the TAIR computer code are shown in figures 24 and 25. Figure 24 shows a pressure coefficient comparison with experiment taken from reference 111. The airfoil is the supercritical CAST 7 and the Mach number is 0.7. The agreement is quite good. Figure 25 shows Mach-number contours around an NACA 0012 airfoil immersed in a 0.95 Mach free stream at an angle of attack of 4° (taken from Holst, ref. 130). The Mach number contours clearly illustrate the existence of a so-called "fishtail" shock-wave pattern downstream of the airfoil trailing edge. This difficult calculation demonstrates the convergence reliability associated with the AF2 transonic relaxation procedure.

The AF2 scheme has been implemented in a number of different applications, including airfoil calculations by Holst (ref. 34) and Atta (ref. 131); cascade flows by Kwak (ref. 132); wing geometries by Holst (ref. 130) and Holst and Thomas (ref. 133); and in wing/pylon/nacelle calculations by Atta and Vadyak (ref. 134). In addition, two other factorizations similar to the AF2 scheme, have been presented by Benek et al. (ref. 135). These two formulations are called AFZ2 and AFZ and are both used for three-dimensional transonic wing calculations. The AFZ2 scheme is very similar to the standard AF2 scheme used in reference 130. The AFZ scheme is a simplification of the AFZ2 scheme which inverts matrices along only the wraparound and normal-like directions, not the span direction. Thus, this scheme is implicit in only two directions, whereas the standard AF2 scheme is implicit along all three coordinate directions. Despite this feature, the simpler AFZ scheme has almost the same convergence properties as the AFZ2 scheme. Numerical results using the three-dimensional AF2 scheme of reference 133 will be presented in section 8.

7.3 Convergence Characteristics of SLOR, ADI, and AF2 Iteration Algorithms

Numerical results comparing the convergence characteristics of the two fully implicit algorithms just presented (ADI and AF2) with the SLOR algorithm are now presented. All three iteration schemes have been applied to the same artificial-density, spatial-differencing scheme for the conservative form of the full-potential equation. A two-dimensional, 10%-thick, circular-arc airfoil with small-disturbance boundary conditions is used as a test case. The finite-difference grid is Cartesian with variable spacing in both the x and y directions. Both subcritical and supercritical cases are considered (M_∞ = 0.7 and 0.84, respectively). Pressure coefficient distributions for these two cases are displayed in figure 26. Note the perfect symmetry associated with the subcritical case and the existence of a moderate strength shock at about 80% of chord for the supercritical case. For more details about these calculations see reference 83.

Convergence characteristics for the subcritical case are displayed in figure 27. All of the convergence parameters for each scheme have been selected by a trial-and-error optimization process. Based on a six-order-of-magnitude reduction in the maximum residual, the ADI scheme is about twice as fast as the AF2 scheme and about 16 times faster than SLOR, in terms of iteration count. However, the ADI and AF2 schemes take about 50% and 30% more CPU time per iteration, respectively, than SLOR; this should be considered when speed ratios based on the total amount of computational work are desired.

Convergence characteristics for the supercritical case are displayed in figure 28. Again, the convergence parameters have been optimized by a trial-and-error process. Based on a six-order-of-magnitude reduction in the maximum residual and in terms of iteration count, AF2 is slightly more than twice as fast as ADI, and about 11 times faster than SLOR. The number of supersonic points (NSP) plotted versus iteration number for the supercritical case is shown in figure 29. The AF2, ADI, and SLOR schemes reach the final value of NSP in 29, 103, and 320 iterations, respectively.

The AF2 iteration scheme was relatively consistent in terms of convergence speed for both cases. The ADI iteration scheme, on the other hand, displayed remarkable speed for the subcritical case but was a disappointment for the supersonic case. This is because the ϕ_{st}-type error term produced by the AF2 factorization is more suitable for supersonic regions than the ϕ_t-type error term resulting from the ADI factorization. In fact, the ϕ_t-type error term has been shown to be destabilizing in the supersonic region (see ref. 29).

Relative levels of convergence for AF2 and SLOR for given reductions in maximum residual are compared in figure 30. The solid lines represent the final solidly converged solution. The other results represent intermediate AF2 and SLOR solutions in which the maximum residual has been reduced by one, two, and three orders of magnitude (figs. 30(a), 30(b), and 30(c), respectively). It is immediately obvious that reducing the maximum residual by equal amounts for the AF2 and SLOR schemes does not produce intermediate results with the same level of error. This behavior can also be observed by comparing the maximum residual history curves of figure 28 with the rms error history curves given in figure 31. The rms error is computed from a formula similar to equation (7.29). The SLOR residual drops very rapidly initially and then levels off. The SLOR rms error drops gradually. Therefore, at the "knee" in the SLOR residual history curve, even though the residual has dropped by about three orders of magnitude, the actual rms error has dropped by only one order of magnitude. In contrast, both maximum residual and rms error results for the ADI and AF2 schemes are nearly straight lines with about the same slope.

This behavior is the result of two factors (see refs. 83 and 129): (1) the AF2 scheme treats all error components equally well (approximately), whereas the SLOR scheme performs efficiently on only the high-frequency error components; and (2) it can be shown that the residual is a weighted sum of errors, in which the weighting factors are the eigenvalues of the finite-difference scheme. The eigenvalue for high-frequency errors is $O(\Delta x^{-2})$; for the low-frequency errors it is $O(1)$. Hence, the residual is heavily influenced by the high-frequency errors. Therefore, the maximum residual operator should not be used as the basis for comparing convergence performance between iteration schemes with different characteristics (for example, AF and SLOR schemes). The rms error is much better suited for this purpose. In practice, using the maximum residual to monitor convergence for either AF or SLOR is the most convenient method (since error is unknown). However, the convergence criterion based on residual should be adjusted (by experience) in accordance with the solution procedure in use.

7.4 AF3 Approximation-Factorization Scheme

Another interesting approximate-factorization scheme, introduced by Baker (ref. 136) for solving the nonconservative full-potential equation is now discussed. The AF3 scheme is given by

$$(-\alpha C\overrightarrow{\delta}_y - A\delta_{xx})(\alpha + \overleftarrow{\delta}_y)C^n_{i,j} = \alpha\omega L\phi^n_{i,j} \tag{7.30}$$

for subsonic regions of flow and by

$$(-\alpha C_c\overrightarrow{\delta}_y - A_c\delta_{xx} - A_u\overleftarrow{\delta}_x)(\alpha + \overleftarrow{\delta}_y)C^n_{i,j} = \alpha\omega L\phi^n_{i,j} \tag{7.31}$$

for supersonic regions of flow (when $u > 0$). The difference direction on the third term of the first factor is reversed when $u < 0$. The coefficients A, C, A_c, C_c, and A_u are determined from the nonconservative full-potential equation written in canonical form (stream and stream-normal coordinates), and are given by

$$\left.\begin{aligned} A_u &= \left(1 - \frac{q^2}{a^2}\right)\frac{u^2}{q^2}, \qquad & A_c &= \frac{v^2}{q^2} \\ C_u &= \left(1 - \frac{q^2}{a^2}\right)\frac{v^2}{q^2}, \qquad & C_c &= \frac{u^2}{q^2} \\ A &= A_u + A_c, \qquad & C &= C_u + C_c \end{aligned}\right\} \tag{7.32}$$

where u and v are the velocity components along the x and y directions, respectively; a is the local speed of sound; and q is the magnitude of the velocity vector. As in the previous AF2 scheme, ω is a relaxation factor,

α an acceleration parameter (cycled over a sequence of values), and $L\phi_{i,j}^n$ is the nth iteration residual defined by

$$L\phi_{i,j}^n = (A\delta_{xx} + B\mu_x\mu_y\delta_x\delta_y + C\delta_{yy})\phi_{i,j}^n \qquad (7.33)$$

for subsonic regions of flow and by

$$L\phi_{i,j}^n = A_c\delta_{xx} + A_u\overleftarrow{\delta}_x\overleftarrow{\delta}_x + B_c\mu_x\mu_y\delta_x\delta_y + B_u\overleftarrow{\delta}_x\overleftarrow{\delta}_y + C_c\delta_{yy} + C_u\overleftarrow{\delta}_y\overleftarrow{\delta}_y)\phi_{i,j}^n \qquad (7.34)$$

for supersonic regions of flow (when $u > 0$ and $v > 0$). The additional coefficients B_u and B_c are given by

$$B_u = 2\left(1 - \frac{q^2}{a^2}\right)\frac{uv}{q}\,, \qquad B_c = -\frac{2uv}{q} \qquad (7.35)$$

The AF3 factorization is similar to the AF2 factorization and produces exceptional improvements in computational speed. This fact is illustrated in figures 32 and 33 (taken from ref. 6). Figure 32 shows the pressure coefficient distribution about an NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2°$ for several different levels of convergence. After just 10 AF iterations, which correspond to about 13 SLOR iterations, the solution is nearly converged. The AF3 convergence history for this case is compared with an SLOR convergence history in figure 33. The SLOR convergence history is enhanced by using the standard grid refinement procedure involving two grids, one coarse and one fine. For this case, the SLOR scheme required over 400 iterations to reach plottable accuracy; the AF3 scheme reached plottable accuracy in about 20 iterations (26 equivalent SLOR iterations).

The two-dimensional AF3 scheme just discussed has been extended to three dimensions by Baker and Forsey (ref. 137). Solutions of the nonconservative full-potential equation have been obtained for wing and wing/fuselage combinations with a factor of 4 or 5 increase in computational efficiency relative to standard SLOR schemes.

A theoretical analysis for various AF schemes, including ADI, AF2, and AF3, is presented by Catherall (ref. 138). In this study, an improvement in computational efficiency is obtained when the contributions of the transformation metrics are properly split between the two factors. In addition, optimal values for the acceleration parameter sequence α and the relaxation factor ω are derived and discussed.

7.5 Multigrid Iteration Schemes

The multigrid iteration scheme is enjoying a wave of popularity that has included applications in a host of different areas. This scheme is actually a convergence acceleration technique and requires a base iteration scheme, for example, SOR, SLOR, or AF. Multigrid schemes have existed for quite some time, having been first introduced by Fedorenko (ref. 139) in 1964. Since then, several authors have analyzed the technique, including Bakhvalov (ref. 140) and, more recently, Nicolaides (ref. 141) and Hackbusch (ref. 142). The most significant aspect of the multigrid iteration scheme is fast convergence. This fast convergence is produced by using a sequence of grids ranging from very coarse to very fine. Each grid is used to eliminate one small range of errors in the error frequency spectrum, namely the errors of highest frequency supported on each mesh. Many relaxation schemes exist that work very well on high-frequency errors, for example, point-Jacobi and AF schemes (with properly chosen acceleration parameters). One of these relaxation schemes is used on each mesh to remove the high-frequency error. A desirable aspect of this approach is that the high-frequency error on the coarsest mesh is actually the lowest frequency error existing in the problem. Because this usually troublesome low-frequency error is efficiently dealt with on a coarse mesh, very little computational work is expended in removing it from the solution. Thus, a tremendous convergence rate enhancement is obtained.

Implementation of a typical multigrid scheme is described in general terms as follows: Suppose we desire a solution to

$$L^h \phi = f \tag{7.36}$$

where L^h is a typical linear difference operator which approximates a differential operator L on a mesh associated with the grid spacing h. The quantity f contains the problem boundary conditions. Let

$$\phi = u + v \tag{7.37}$$

where u is an approximation to ϕ and v represents an error. Therefore, as the iteration scheme converges, $u \rightarrow \phi$ and $v \rightarrow 0$. The basic multigrid scheme can be expressed by

$$L^{2h} v + I_h^{2h}(L^h u - f) = 0 \tag{7.38}$$

where L^{2h} is a finite-difference operator which approximates L on a mesh associated with the grid spacing 2h, instead of h, that is, twice as coarse as the original mesh. The operator I_h^{2h} is an interpolation or averaging operator which transfers values of the residual $(L^h u - f)$ from the fine mesh to the coarse mesh. After the coarse mesh corrections, v, are obtained, they are transferred back to the fine mesh by using

$$u^{new} = u + I_{2h}^h v \tag{7.39}$$

where I_{2h}^h is an interpolation operator. The process can continue to coarser meshes so that ultimately just one or maybe several mesh cell widths span the entire domain of interest.

To extend the idea to nonlinear problems a simple modification is helpful. By adding and subtracting $L^{2h}u$ from equation (7.38) the new form becomes

$$L^{2h} \bar{u} = \bar{f} \tag{7.40}$$

where

$$\left.\begin{aligned} \bar{u} &= u + v \\ \bar{f} &= L^{2h} u - I_h^{2h}(L^h u - f) \end{aligned}\right\} \tag{7.41}$$

The quantity $\bar{u}$ represents a new or improved estimate of ϕ which is determined from the coarse mesh. The quantity $\bar{f}$ is a modified right-hand side which essentially represents the difference in residuals between the h and 2h meshes. New updated coarse values are transferred back to the fine mesh by using

$$u^{new} = u + I^h_{2h}(\bar{u} - u) \qquad (7.42)$$

Thus, the error quantity v does not have to be stored as in the original version.

Applications utilizing the multigrid scheme were slow to materialize after its introduction, primarily because of difficulties in implementation and general underestimation of the potential of multigrid enhanced schemes. The first work to apply the multigrid scheme numerically was that of Brandt (ref. 143) in 1972. Later, the multigrid scheme was formulated in general terms by Brandt (ref. 144). In this latter reference, a good historical background of the multigrid scheme is presented, including a review of related earlier work.

The first use of the multigrid scheme for transonic calculations was presented by South and Brandt (ref. 145). In that study, numerical solutions of the TSD equation for nonlifting airfoils were obtained. The speed of an optimized SLOR scheme was improved by a factor of 5 on uniform meshes and by a factor of 2 on stretched meshes. A primary difficulty reported by South and Brandt involved the existence of a variety of limit-cycle oscillations between several grids, thus inhibiting convergence. This problem seemed to be the result of insufficient smoothing of the high-frequency errors on one grid before passing to the next coarser grid. South and Brandt concluded that the SLOR base algorithm used in the multigrid scheme did not have uniform smoothing properties in both directions, especially for nonuniform, highly stretched meshes. They hypothesized that alternating the SLOR sweep direction or utilizing an ADI iteration scheme as the base algorithm might solve this problem.

Another approach, proposed by Arlinger (ref. 146), is to refine or coarsen the mesh in only one coordinate direction while doing line relaxation along the opposite direction. This technique produced a convergence rate acceleration but did not take full advantage of the multigrid scheme. To date, the most successful application of a multigrid convergence acceleration scheme to a practical transonic problem is the work of Jameson (ref. 147).

In that study, the full-potential equation in conservative form is solved, using a multigrid scheme with a specially constructed AF base iteration scheme. This scheme, when applied to the following linear model equation,

$$A\phi_{xx} + B\phi_{yy} = 0 \qquad (7.43)$$

is given by

$$(S - A\delta_{xx})(S - B\delta_{yy})C^n_{i,j} = \omega SL\phi^n_{i,j} \qquad (7.44)$$

where A and B are constants, ω is the standard relaxation factor, and S and L are operators defined by

$$S = \alpha_o + \alpha_1\delta^-_x + \alpha_2\delta^-_y \qquad (7.45)$$

and

$$L\phi^n_{i,j} = (A\delta_{xx} + B\delta_{yy})\phi^n_{i,j} \qquad (7.46)$$

In equation (7.45) α_0, α_1, and α_2 are parameters that depend on flow type and user input. The quantities δ_x^- and δ_y^- denote first-order-accurate upwind difference operators in the x and y directions, respectively.

The Jameson scheme uses a recursive approach for implementing the multigrid philosophy, instead of the adaptive approach advocated by Brandt (refs. 143 and 144). In the adaptive multigrid approach, the decision to proceed to the next mesh, either coarser or finer, is based on a convergence rate criterion. If the solution residual is dropping slowly, the iteration process proceeds to coarser meshes. Conversely, if the solution residual is dropping rapidly, the iteration proceeds to finer meshes. In the recursive approach of Jameson, a single multigrid cycle starts with an AF iteration on the finest mesh, followed by an AF iteration on the second finest mesh, etc. This continues until the coarsest mesh is reached. Then the process is reversed, starting with the coarsest mesh and ending with the second finest mesh. Therefore, each multigrid grid cycle consists of one AF application on the finest mesh and two applications on each of the remaining meshes. If a fine grid AF iteration is defined as a unit of work, then one multigrid cycle, using the recursive approach, requires about 1-2/3 work units plus interpolation operations.

Results produced by the Jameson multigrid scheme are displayed in figures 34 and 35. The pressure coefficient distribution for an NACA 64A410 airfoil at a free-stream Mach number of 0.72 and at an angle of attack of 0° is displayed in figure 34. A moderate-strength shock exists at about 60% of chord. Notice that the residual has been reduced below 10E-12 (see fig. 34), which is approximately an eight-order-of-magnitude reduction from the initial value; the reduction was achieved in only 29 multigrid cycles. Convergence histories for this case, which were computed using different numbers of meshes (from one mesh, that is, no multigrid, up to five meshes), are shown in figure 35. The convergence rate (CR), which is defined as the mean reduction in the average residual per unit of work, is also displayed for each curve. Increasing the number of meshes or, equivalently, increasing the coarseness of the coarsest mesh, greatly improves the convergence rate.

Other researchers have used the multigrid algorithm to sove the full-potential equation in a variety of applications: Fuchs (ref. 148) and Deconinck and Hirsch (ref. 150), for two-dimensional applications; Arlinger (ref. 150), for axisymmetric calculations; McCarthy and Rehner (ref. 151) and Brown (ref. 152), for three-dimensional engine-inlet calculations; and Shmilovich and Caughey (ref. 153) and Caughey (ref. 154), for three-dimensional wing calculations.

7.6 Other Iteration Schemes

The strongly implicit procedure (SIP) introduced by Stone (ref. 155) has been applied to the numerical solution of the full-potential equation for transonic airfoil calculations by Sankar and Tassa (ref. 156). Additional applications include those of Sankar et al. (ref. 157), for steady transonic wing calculations, and Roach and Sankar (ref. 158), for transonic cascade calculations. In all cases the SIP solution algorithm displayed good convergence characteristics as a relaxation scheme. In addition, the SIP algorithm has the ability to compute time-accurate flow fields; see Sankar et al. (ref. 159) for unsteady wing calculations.

The SIP iteration scheme requires three additional arrays of storage (five arrays for the SIP scheme, two for the ADI or AF2 schemes) and requires a few more operations to invert the resulting matrix equations. The SIP method has a built-in mechanism for matrix conditioning. That is, the scheme automatically adjusts the matrix diagonal entries so that the matrix is well conditioned, even in regions where the mesh spacing becomes coarse. Additional properties regarding the SIP iteration scheme are discussed in reference 157.

Several researchers have presented new algorithms especially designed for vector computers, including Keller and Jameson (ref. 160), Hafez et al. (ref. 84), Redhead et al. (ref. 161), Hotovy and Dickson (ref. 162), and South et al. (ref. 87). Vector computers offer greatly enhanced computing speeds arising from the ability to operate on many calculations simultaneously (parallel machines) or in an assembly line fashion (pipeline machines). [See Bailey (ref. 163) for more discussion of computer architectures.] In references 84, 160, and 162 the algorithm's vector characteristics were stressed above all else. As a result, these vector algorithms could process very large numbers of grid points per second but usually required more iterations than standard SLOR to converge.

In South et al. (ref. 87) an algorithm called Zebra II, which is highly vectorizable and requires about the same number of iterations to converge as does SLOR, is described. This algorithm is an explicit, or point, scheme which mimicks a full-plane SOR algorithm and is designed to solve the conservative full-potential equation.

The Zebra II algorithm takes a step in the right direction, but other approaches for vector computation may still be superior. A theoretical study comparing the vector processing attributes of four transonic full-potential algorithms (SLOR, ADI, AF2, and Zebra II) was performed by Holst (ref. 164) utilizing a mathematical model of a pipeline vector computer. The results of this study indicate that implicit algorithms, which contain nonvectorizing matrix inversions, still enjoy an overall supremacy on vector computers relative to explicit, or point, iterative techniques, when all aspects of efficiency are taken into consideration. Results from a vectorization study performed on a three-dimensional transonic wing code will be presented in the next section.

Other iteration schemes suitable for producing fast convergence for the full-potential equation include the extrapolation schemes of Hafez and Cheng (ref. 165), Caughey and Jameson (ref. 49), and Yu and Rubbert (ref. 166); the conjugate-gradient methods of Bristeau et al. (ref. 119), Glowinski et al. (ref. 167), Chattot and Coulombeix (ref. 168), and Wong and Hafez (refs. 169 and 170); and the minimum residual method of Wong and Hafez (ref. 171). The work of Wong and Hafez (ref. 169) provides an interesting discussion of iteration schemes for solving the full-potential equation. Results are presented for several schemes, including SLOR, two variations of the ZEBRA scheme mentioned above, and conjugate-gradient schemes with several types of preconditioning combined with both SLOR and ZEBRA schemes. These schemes are applied to two spatial discretization schemes, including a finite-difference scheme and a finite-element scheme. It is found that the combined iteration schemes are superior to the standard SLOR scheme by as much as a factor of 10 for subcritical cases and by at least a factor of 2 for tough transonic cases. Details of each of these schemes can be found in the references cited above.

The multiline, or block, iterative schemes presented in Hafez and Lovell (ref. 172) represent another competitive form of relaxation algorithm. In this type of scheme two or three lines of the grid are treated in the same matrix inversion. Thus, the amount of implicitness normally associated with a single line inversion scheme such as SLOR, is greatly increased. Several convergence rate comparisons are presented in reference 172 and indicate competitive computational efficiencies for the multiline schemes relative to other types of iteration schemes. In addition, with red-black ordering of the blocks, these schemes are easily vectorized.

8. APPLICATIONS IN THREE DIMENSIONS

Extension of the ideas presented in the previous chapters to three dimensions, for example, wing and wing/fuselage calculations, is very much complicated by storage and computer time requirements. Nevertheless, the benefits to be obtained are large and much work has been done in this area. In this section a brief review of existing techniques including specific aspects of a number of widely used three-dimensional transonic computer codes is presented. For a more detailed review of this and other related subjects, see reference 2.

8.1 Computer Code Characteristics

Several computer codes designed to solve the three-dimensional transonic flow over wing and wing/fuselage configurations have been developed and are currently in wide use in the aircraft industry. A few of these codes are listed in table 2 along with specific references describing the details of each code. Note that both conservative and nonconservative forms of the full-potential formulation are represented. More discussion concerning the various transonic potential formulations, including both full-potential and TSD forms, is given in chapter 2 of reference 2.

The iteration schemes used by the codes listed in table 2 are predominantly SLOR. The newer approximate-factorization or multigrid schemes are utilized in only the more recent codes. This produces very slow and thus costly convergence for most of the older codes and makes them relatively expensive to run. One feature generally used in SLOR codes to improve computational efficiency is the use of grid refinement, that is, the use of a coarse-medium-fine grid sequence to accelerate convergence. Converged results from the coarse mesh are interpolated onto the medium mesh, then reconverged and finally interpolated from the medium mesh to the final fine mesh. Thus, a good initial solution is provided for the fine-mesh calculation. Use of this grid sequence philosophy increases the computational efficiency by a factor of 2 or 3 (at least for crude levels of convergence).

The overall computer time required for a complete transonic wing solution varies greatly, depending on the level of convergence desired, the number of grid points used, and the type of formulation chosen. Even the amount of computer time reported by two different users from the same computer code can be different by factors of 3-5, simply because of the level of convergence desired and the number of grid points used. Computer time comparisons which address most of these aspects are given in table 3 (taken from Hinson and Burdges (ref. 185). Note that all codes compared have about the same number of wing-surface grid points, whereas the total number of field points is quite different. The Bailey-Ballhaus code (see ref. 186) uses a concept called grid embedding (first introduced by Boppe (ref. 187) to more adequately cluster grid points at the wing surface). This concept is very attractive because the resulting mesh topology is much more efficient and the wing-surface results do not seem to be affected by this treatment. Grid embedding has not been used extensively with the full-potential formulation, probably because of the resulting mapping complications. In addition, full-potential grids are generally more efficient than TSD grids (that is, TSD grids without grid embedding), and therefore, grid embedding would not be as useful in the full-potential formulation. (See Brown, ref. 188, for one application using the full-potential equation and embedded grids.)

All three results shown in table 3 were obtained for the same transonic wing at identical test conditions. All three codes used a coarse-medium-fine sequence. Convergence was monitored by examining the pressure distribution histories at two span locations. The Bailey-Ballhaus TSD code converged in approximately 200 iterations, the FL022 code in about 50-100 iterations, and the FL027 code in about 200 iterations. For these conditions, the FL027 code solution time is approximately twice that of the FL022 code and about 5 times that of the Bailey-Ballhaus code. The convergence characteristics were essentially the same for both the conservative and nonconservative options available in the Bailey-Ballhaus code. For more information about this set of comparisons see reference 185.

Another computer time comparison is reproduced in table 4 from reference 183. In this comparison, results from the FL028 computer code on the CDC 7600 computer and the TWING computer code on both the CDC 7600 computer and the new Cray-1S vector computer are presented. The recently introduced TWING computer code is similar to the FL028 code in that it solves the three-dimensional full-potential equation in conservative form, but different in that it uses the fully implicit AF2 iteration scheme (see sec. 7.2).

The results for both codes presented in table 4 have been obtained from the same problem (ONERA M6 wing, $M_\infty = 0.84$, $\alpha = 3.06°$). The convergence parameters from both codes have been approximately optimized by a trial-and-error process. The TWING results are based on an 83-iteration run in which the center-span lift changed by 0.03% in the last 30 iterations. The FL028 results are based on a coarse-medium-fine mesh sequence with 50, 50, and 283 iterations, respectively. The total lift for this run changed by 0.49% in the last 100 iterations of the fine mesh. Because FL028 uses a less efficient mapping, more field points are required to achieve approximately the same number of surface grid points used in the TWING code.

Immediately obvious from table 4 is that TWING sets up the lift much faster than FL028 (14 times faster for 98% of the lift) when both codes are run on the same computer (CDC 7600). Of course, this improvement is due to the advanced fully implicit iteration scheme utilized by TWING (see sec. 7.2 for more discussion of this point). In addition, the vectorized version of TWING run on the Cray-1S vector computer is about 11 times faster than TWING run on the CDC 7600 computer. These run times are for the flow-solver portion of both codes only and do not include times for grid generation, initialization, or solution printout.

The speed improvement offered by TWING is, of course, largely a result of the faster computer hardware associated with the Cray-1S computer. But another important reason for the improvement is that the AF2 iteration scheme in the TWING code is highly vectorizable. All the interior grid point operations performed by the TWING code vectorize, including all operations associated with the matrix inversions of each sweep, the upwinding logic associated with the density coefficients, and the logic associated with the $\phi_{\xi t}$ time-like damping term. Despite this complete vectorization, the code is almost completely written in FORTRAN and will run on any standard computer with a FORTRAN compiler. The FL028 computer program would benefit from the vectorization offered by the Cray computer, but would have difficulty obtaining the level of efficiency achieved by the TWING computer code. This is because a portion of the SLOR algorithm used within FL028 is inherently recursive, and, therefore, not vectorizable. For more information regarding these results see reference 183.

The Jameson-Caughey nonconservative full-potential computer code (FL022) uses a sheared parabolic coordinate system which can be defined by (see ref. 173)

$$\left.\begin{aligned} x_1 + iy_1 &= \{x - x_o(z) + i[y - y_o(z)]\}^{1/2} \\ z_1 &= z \end{aligned}\right\} \qquad (8.1)$$

where z is the spanwise coordinate, and x_o and y_o define a singular line of the coordinate system located just inside the leading edge. The effect of this transformation is to unwrap the wing to form a shallow bump (see fig. 36):

$$y_1 = s(x_1,y_1) \qquad (8.2)$$

Next, a shearing transformation is used

$$\left.\begin{aligned} \xi &= x \\ \eta &= y_1 - s(x_1,y_1) \\ \zeta &= z \end{aligned}\right\} \qquad (8.3)$$

to map the wing surface to a coordinate surface. Finally, the ξ, η, and ζ coordinates are stretched with suitable stretching formulas. A more revealing diagram of the overall wing grid system shown in the physical domain is reproduced from reference 189 in figure 37. Note that the airfoil cross-sectional grid produced by this transformation technique is of the "C" mesh variety.

An interesting aspect of the FLO22 algorithm, created by the grid topology, is the orientation of the tridiagonal matrices used in the SLOR iteration scheme. This aspect is illustrated in figure 38, which has been taken from reference 173. This diagram shows the computational domain in a spanwise cross-sectional plane; note the orientation of the streamlines. The purpose of sweeping through the mesh in this fashion is to always avoid sweeping in an upwind direction in the supersonic region. If this procedure is not followed, unstable operation could result.

The Jameson-Caughey FLO27 computer code is capable of treating the flow about isolated swept wings or wings mounted on an infinite cylinder. The FLO28 code is a closely related derivative of FLO27 which allows more sophisticated treatment of the fuselage. The FLO30 code allows a still more sophisticated treatment of the wing/fuselage problem. The relative level of sophistication of each of these formulations is displayed in figure 39 (taken from ref. 190). The actual transformation used for most FLO27 and FLO28 wing calculations is very similar to the transformations described for the FLO22 computer program and will not be discussed further.

The ONERA transonic full-potential code of Chattot et al. (ref. 107) is capable of treating the flow about isolated wing geometries. The mapping used is similar to the previous descriptions and will not be discussed further. This code, like the more recent FLO codes (FLO27, FLO28, and FLO30), uses numerical evaluations of the metric quantities and, therefore, could support mesh generation routines of varying types provided they all used the same general topology.

Most full-potential codes are designed for either conservative or nonconservative spatial differencing schemes. However, the ONERA code includes options for both forms. This allows direct comparison of the two forms in a format that is unencumbered by different iteration algorithms, grid topologies, or programming styles. Such a comparison is shown in figure 40 for a rectangular planform, nonlifting NACA 0012 wing at a free-stream Mach number of 0.85. The conservative result produces a shock that is too strong and about 10%-15% downstream of the experimental shock position. The nonconservative shock location is incorrect in just the opposite direction; it is too weak and too far forward. This type of disagreement between conservative and nonconservative forms is characteristic. The code user should be aware of which formulation is being used and how the resulting solutions should be interpreted.

The ARA full-potential wing/body code of Forsey and Car (ref. 178), is capable of solving the nonconservative full-potential equation about isolated wings and wing/body combinations. Bodies must be circular in cross-section although the body radius may vary axially. This has the effect of providing a relatively good wind/body interference simulation but does not produce good finite-length body effects.

The mapping procedure used in the ARA code is implemented in two phases. First, the body is transformed via conformal mapping into a slit that corresponds to a portion of the symmetry plane. Because each body cross section is constrained to be circular, this mapping is analytic. Next, the wing leading and trailing edges are extended to infinity to produce a "flat-plate wing" outboard of the tip. Each flat-plate wing cross section is mapped to the interior of a circle using an analytic conformal mapping, and each finite-thickness wing cross section is mapped using a numerical conformal mapping. Thus, the total effect of this transformation is to map the surface of the wing (including the flat-plate wing extension) to the inside surface of a circular cylinder and the free-stream outer boundary to the circular cylinder center. Unlike the FLO full-potential codes (F1022, FL027, FL028, and FL030), which use a "C" mesh topology about each wing span station, the ARA code uses an "O" mesh topology. This produces a somewhat more efficient mesh for the ARA code; that is, for the same number of total grid points, the ARA code mesh puts about 30% to 40% more points on the wing surface than a typical FLO code mesh.

The TWING code is capable of treating isolated-wing or simplified wing/fuselage geometries. The finite-difference grid is generated by an elliptic-solver numerical grid-generation procedure; it is described in Holst and Thomas (ref. 133). An example grid generated using this procedure is displayed in figure 41 (taken from ref. 191). The geometry used is a highly swept, low-aspect-ratio wing with a large amount of twist (Hinson-Burdges Wing C, see ref. 185). Figure 41(a) shows the entire planform view of the wing. Three airfoil cross-section plots of the grid are displayed in figures 41(b)-41(d): (1) at the wing root, (2) just inboard of the tip, and (3) just outboard of the tip, respectively. The exact positions of these cross-sectional plots are shown in the planform view of figure 41(a). The x/c and z/c Cartesian coordinates displayed in figures 41(b)-41(d) are plotted to the same scale and are normalized by the root chord, c. Note the large amount of twist and the efficiency with which the "O" mesh topology clusters grid points about each airfoil cross section.

Finally, figure 41(e) shows a perspective view of the Wing C grid as generated by the TWING computer code. The symmetry plane, wing surface, and vortex sheet grids are all displayed. For clarity, only every fourth grid line in the wraparound direction is plotted. This view (except for the trailing-edge vortex sheet) corresponds very closely to the view of the actual wing mounted in the wind tunnel.

8.2 Results Obtained with Existing Codes

The next item of importance is accuracy. That is, how well do these codes actually predict transonic-wing, flow-field physics? In this section, computed results collected from a variety of sources will be displayed in an attempt to quantify the answer to this question.

A series of results taken from reference 107 is shown in figure 42. Three numerical results, Jameson (conservative), Bailey/Ballhaus (TSD conservative, see ref. 186), and ONERA (nonconservative) are compared with experiment at a wing semispan station of 20%. The configuration used for this set of calculations is the ONERA M6 wing at $M_\infty = 0.841$, $\alpha = 3°$. (See ref. 192 for more information on the experimental results and geometrical characteristics of the ONERA M6 wing.) This frequently used ONERA M6 test case represents a difficult test for a transonic flow calculation procedure because of the existence of a double shock. The double-shock structure is preducted by all three results with some variation in the second shock position. The conservative results predict a somewhat stronger second shock slightly downstream of the nonconservative shock.

Results from the ARA full-potential code (taken from ref. 2) are presented in figures 43 and 44. The wing/body configuration used in this calculation is shown in figure 43. The body was a circular cylinder of constant radius, and the mid-mounted wing had the following characteristics: AR = 6, TR = 1, sweep = 25°. The wing had a constant supercritical airfoil section and was used without twist. Viscous effects were simulated by the addition of a displacement thickness obtained from a two-dimensional transonic viscous code (see ref. 193).

Computed results for a free-stream Mach number of 0.86 and an angle of attack of 4.2° are compared, in figure 44, with experimental results for the same Mach number and two angles of attack, 4.6° and 3.7°. The calculation was performed with the usual grid sequence involving 300, 100, and 160 iterations on the coarse, medium, and fine meshes, respectively. The finest mesh for this case consisted of 160 × 20 × 24 = 76,800 grid points. The computational time required for this calculation was equivalent to about 25 min of CPU time on the CDC 7600 computer. Results are shown for three span stations ($y = 0.37$, 0.55, and 0.73), and are generally in excellent agreement with experiment. Note the unusually sharp shock capture displayed in figures 44(b) and 44(c). This is a result of the efficient mesh topology used in the ARA full-potential code in conjunction with a relatively fine mesh.

Additional results computed from the ARA full-potential computer code are displayed in figures 45-47 (taken from ref. 137). Figure 45 shows the wing/fuselage geometry used in this calculation. For more details regarding this configuration, which has been designated as an AGARD test case for three-dimensional method evaluation, see reference 194. Pressure distributions for this geometry at $M_\infty = 0.9$ and $\alpha = 1°$ are displayed in figure 46. These results were computed on a relatively coarse mesh consisting of 80 × 24 × 10 = 19,200 points. The agreement between the computed and experimental results is good at all three semispan stations shown in figure 46.

The results just presented have been computed with the new version of the ARA code which uses the AF3 approximate-factorization scheme of Baker and Forsey (ref. 137). Lift development versus the equivalent number of relaxation iterations is presented for this case in figure 47. Four curves are displayed corresponding to AF and SLOR iteration schemes applied to the wing/body configuration of figure 45 and a wing-alone variation of the same configuration. The AF3 convergence rate displayed in figure 47 is far superior to the SLOR convergence rate for both the wing-alone and the wing/fuselage cases. The SLOR scheme used in figure 47 did not use the standard grid sequence which would have improved convergence. However, the AF3 scheme still possess a large advantage over the SLOR scheme in convergence speed.

Pressure comparisons for the FL022 and FL027 computer codes are presented in figures 48 and 49 for a high-aspect-ratio (AR = 10.3) supercritical wing tested by NASA Langley (taken from ref. 195). The wing is swept 27° at the quarter chord line, and has streamwise section thickness ratios of 14.9% at the fuselage junction, 12% at the trailing-edge break, and 10.6% at the tip. (See ref. 196 for a more detailed description of the geometry and experimental results.) Viscous effects were modeled by a two-dimensional integral boundary-layer method applied in streamwise strips. The calculation angles of attack were determined by matching the experimental lift coefficients.

Figure 48 presents results from FL022; FL022 does not model the fuselage used in the experiment. In an attempt to account for the fuselage interference effect, this calculation was run at a free-stream Mach number of 0.80, whereas the experiment was run at 0.79. Agreement is good everywhere except near the inboard stations, where details of the fuselage interference are probably important.

Figure 49 presents results from FL027 for the same case as that shown in figure 48. For this case, however, the fuselage is modeled as an infinite cylinder with a radius equal to the maximum radius of the fuselage on which the wing was mounted in the experiment. The low-mounted position of the test wing was also simulated in the numerical calculation. For this case, the aeroelastic deformation of the model was estimated by introducing 0.36° of wash out (negative twist) at the tip. This aeroelastic twist was added linearly from the trailing-edge break to the tip; it was not included in the FL022 calculation of figure 48. The Mach-number correction used for the FL027 result was 0.007 and was applied to account for the effect of the finite fuselage used in the experiment. It is clear from figure 49 that representation of the effect of a finite fuselage by a simple Mach-number shift is an oversimplification. The results in figure 49 are in better agreement with experiment near the root station than the results of figure 48, but in poorer agreement outboard near the tip. Nevertheless, reasonable agreement is obtained in both cases by making these ad hoc adjustements or calibrations.

A relatively simple, but adequate way in which to introduce fuselage effects into isolated-wing codes is discussed in Verhoff and O'Neil (ref. 190) and by Van der Vooren et al. (ref. 197). In the latter approach, spanwise velocities are prescribed at the vertical wing-root plane in the FL022 computer code. These velocity components are determined from a panel-method calculation for the complete wing-body configuration. Pressure distributions showing the success of this technique are presented in figure 50. Numerically computed results with and without this root-plane modification are compared with experiment at four semispan stations. In this calculation the fuselage effects are extremely important immediately at the root plane but die out rapidly as midspan is approached.

The next result presented in this section is from the Yu computer code (refs. 181 and 182). This code is basically a modified version of the Jameson-Caughey finite-volume code. The primary improvement in the Yu code is a new efficient and flexible grid-generation method based on the body-fitted coordinate method of Thompson et al. (refs. 36-38). In this approach, a set of nonlinear elliptic equations is solved numerically to determine the three-dimensional grid. The interior grid distributions are controlled by the boundary grid distributions in a manner developed by Thomas and Moddlecoff (ref. 39). For more information about this grid-generation technique see section 5.2.

Results from the Yu code (taken from ref. 2) are presented in figures 51 and 52. The wing/fuselage configuration is shown in figure 51. In this calculation procedure the restriction to simplified fuselage geometries is not required; reasonably complex geometries can be accurately simulated. In this case, the flow field about the Boeing 747-200 wing/fuselage geometry is computed for a free-stream Mach number of 0.84 and an angle of attack of 2.8°. The section pressure coefficient distributions are compared with experiment at three span stations in figure 52. The agreement at each station is excellent.

A comparison of FL028 and TWING results with experiment for the ONERA M6 wing at $M_\infty = 0.84$ and $\alpha = 3.06°$ is presented in figures 53-56 (taken from ref. 133). The finite-difference grid used by TWING for this calculation consisted of 89 × 25 × 18 = 40,050 points (wraparound, spanwise, and normal-like directions, respectively) with 89 × 17 = 1513 points on the wing surface. The FL028 results were computed on a mesh with 120 × 16 × 28 = 53,760 points (wraparound, normal-like and spanwise directions, respectively) with 75 × 18 = 1350 points on the wing surface. The TWING cross-sectional grid uses the "O" mesh topology, and FL028 uses the "C" mesh topology. This produces a somewhat less efficient mesh for FL028 relative to TWING and is the basic reason for comparing solutions computed on different size meshes.

Pressure coefficient comparisons are presented in figure 53 at four different semispan stations (η = 0.20, 0.44, 0.65, and 0.90). The general agreement of the results is good, especially considering the coarseness of the meshes involved. The largest discrepancy arises at the aft shock. The FL028 shock is in better agreement with experiment, being very close or slightly upstream, and the TWING shock is slightly downstream. Since these two codes are conservative inviscid formulations it is expected that the inclusion of viscous effects would move this shock forward. Such a movement would bring the TWING result into better agreement with experiment and have the opposite effect for the FL028 result.

The overall shock sonic-line position for all three results is compared in figure 54. These results correspond to the shock sonic line computed by linear interpolation and do not necessarily reflect the position of the steepest shock pressure gradient. The experimental points almost exactly split the two numerical results, with FL028 being upstream and TWING downstream.

The convergence properties of the TWING computer code for the ONERA M6 wing problem just presented are displayed in figures 55 and 56. The convergence parameters for this case have been approximately optimized by a trial-and-error process. Figure 55 shows the buildup of the number of supersonic points (NSP) and the center-span, section-lift coefficient with iteration number (n). Also plotted along the horizontal axis is the CPU time associated with both the CDC 7600 and Cray 1S computers. (Note that the computer times for this case have already been presented and compared with the FL028 computer times in section 8.1; see table 4). Each symbol plotted (fig. 55) represents one cycle in the α acceleration parameter sequence. At 24 iterations, or three α cycles, the solution is nearly converged. This corresponds to just 4.8 sec of CPU time on the Cray 1S vector computer!

The rate of development of the pressure coefficient distribution at one selected semispan station (η = 0.18) is presented in figure 56. The solution after 20 iterations is compared with the tightly converged solution after 80 iterations. Very little disagreement exists, except at the shock, where about six or eight points have not quite converged to their final steady-state values.

The next result presented is a TWING calculation (taken from ref. 191) for the Wing C configuration described in Hinson and Burdges (ref. 185) (see also ref. 198). This wing geometry is characterized by a small aspect ratio (AR = 2.6), large sweep (Λ_{LE} = 45°), small taper ratio (TR = 0.3), and has a large amount of twist (twist = 8°). The grid for this configuration has already been presented in figure 41; it consists of 127 × 27 × 20 = 68,580 points with 127 × 17 = 2159 points on the wing surface. Pressure distribution comparisons between the TWING results and experiment are shown in figure 57. For this case, the experimental Mach number and angle of attack were 0.85 and 5.9°. The computational Mach number and angle of attack were 0.83 and 5.0°. The numerically computed corrections (ΔM = -0.02 and $\Delta\alpha$ = -0.9°) were determined by trial-and-error; they represent (approximately) the best experimental/numerical correlation for this set of conditions. The set of corrections determined for these data in reference 198 (ΔM = -0.005 and $\Delta\alpha$ = -0.9°) are similar to but smaller than those computed by TWING.

The agreement for this case is quite good everywhere except at the tip, where the need for viscous corrections is apparent. Of particular note in this calculation is the ability of the TWING code to predict the oblique shock, which exists at both the third (y/c = 0.5) and fourth (y/c = 0.7) semispan stations. The differencing scheme in this region is entirely first-order accurate and yet little shock smearing is exhibited.

To obtain an idea of the effects of the angle of attack ($\Delta\alpha$) and the Mach number (ΔM) corrections, several results are shown in figure 58 (taken from Subramanian et al., ref. 191). Pressure distributions at two semispan stations for the Wing C configuration just discussed are compared with experiment for several conditions: (1) the uncorrected experimental conditions (M_∞ = 0.85, α = 5.9°); (2) the corrected experimental conditions, using the corrections cited in reference 198 (M_∞ = 0.845, α = 5.0°); and (3) the corrected experimental conditions, using the corrections computed in reference 191 (M_∞ = 0.83, α = 5.0°). As seen in figure 58, the angle-of-attack correction is more important than the Mach-number correction. The corrections cited in reference 198 yield a reasonable solution in the present case, primarily because both angle-of-attack corrections are the same.

The last set of results is from a version of the FL030 computer code in which a sophisticated viscous correction model has been added by Streett (ref. 199). This model consists of a three-dimensional, integral boundary-layer method based on the work of Stock (ref. 200) and Smith (ref. 201). In addition, a strip wake model accounting for thickness and curvature effects is also included. Results using this computational technique are displayed in figures 59 and 60.

Figure 59 shows a pressure coefficient distribution compared with experiment for the Hinson-Burdges Wing A geometry at M_∞ = 0.819 and α = 1.96° (see ref. 185). Three semispan stations are shown, η = 0.28, 0.50, and 0.68. In this case, the experimental Mach number and angle of attack are matched. Note the excellent agreement with experiment.

Figure 60 shows the lift reduction owing to viscous effects, R, which is defined by

$$\eta = \frac{C_{\ell,\text{inviscid}} - C_\ell}{C_{\ell,\text{inviscid}}} \tag{8.4}$$

plotted versus span station for the advanced transport configuration presented in reference 196. The Mach number and angle of attack for both the experiment and the computation were 0.78 and 1.65°, respectively. The two curves displayed in figure 60 correspond to the full wake model of Streett (ref. 199) and a similar wake model without curvature effects. Viscous effects are very important for this calculation, as shown by both curves in figure 60. In addition, the wake-curvature effect is shown to be substantial outboard of the mid semispan position, especially near the wing tip.

The preceding results are offered in an attempt to answer the question posed at the beginning of this section: How well do these schemes model transonic flow-field physics? In some cases, the modeling is quite good, in other cases not so good. The subjects of geometrical modeling and viscous corrections play the most important roles in obtaining good experimental/numerical correlations. Other effects, such as wind-tunnel-wall interference, tunnel flow angularity, model distortion, numerical truncation error (coarse mesh effects), and lack of solution convergence, also must be considered in the final analysis.

Most of the three-dimensional results presented in this section show at least one new or different attribute; for example, computational efficiency, geometrical generality, or good viscous corrections. In most codes, only one area is fully developed; thus, many codes exist that are, for example, very accurate but very slow or that are very fast but do not contain accurate viscous effects. Consequently, there will have to be additional research in consolidation of techniques before truly useful transonic codes can be made available. The ultimate transonic code must have good characteristics in all respects — accuracy, speed, reliability, and the capability of solving flow fields over a variety of different aircraft configurations.

9. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

The numerical solution of the transonic full-potential equation has received much attention within the CFD research community in the past 10-12 years. The purpose of these notes has been to review selected topics associated with this field of research. Classical relaxation schemes and early ideas associated with transonic potential schemes were touched upon first. Current research acticities were then discussed, with emphasis on grid-generation techniques, spatial differencing schemes, and iteration schemes. Special emphasis was placed on the convergence acceleration characteristics of iteration schemes since this aspect largely controls computational cost. Finally, a series of three-dimensional results from a variety of different sources was presented, thus providing the reader with a good basis for evaluating the state of the art in this field.

One obvious conclusion from this presentation is that the numerical solution of the full-potential equation is highly developed. The aircraft industry utilizes, on a routine basis, numerical solutions of the full-potential formulation to gain insight into the transonic flow fields for many two- and three-dimensional applications. It is projected that with the cost of computations going down and the cost of wind-tunnel tests going up, the use of these computational tools for the design and development of modern aircraft will increase steadily in the future.

Before the application of these schemes can become truly production-oriented, there will have to be improvements in several areas. First, computer codes must be more flexible and, therefore, applicable to a larger range of geometries. Advances in this area are largely paced by developments in surface-geometry representation and grid generation. Second, accuracy must be improved. This includes relatively simple areas including boundary-condition application, metric differencing, and grid-point clustering in regions of large solution gradients, as well as more complicated aspects associated with viscous effects. Third, in the area of viscous effects, the accurate prediction of moderate separation regions must be improved. And fourth, the computational efficiency and reliability of these schemes should be improved. That is, application of a specific technique to a slightly different geometry or new set of free-stream conditions should not result in divergence or a significant loss of speed.

As indicated in the last section, significant progress has been made in many of these areas. However, most of the computer codes in existence today have serious failings in one or more of the areas mentioned above. Therefore, a clear-cut line of future research is in consolidating the advances that already exist. The ultimate computer code must contain all of the characteristics mentioned in the preceding paragraph: generality, accuracy, reliability, and computational efficiency.

This type of research is more atuned to applications rather than basic algorithm development, but research associated with basic algorithm development is also required. Many of these topics are associated with grid-generation algorithms. Interface differencing procedures associated with overlapping grid systems or component adaptive grids need to be investigated. This approach seems to be the best way (at least for finite-difference discretization schemes) of eventually producing transonic flow solutions about complicated geometries — for example, over a complete aircraft. Other research opportunities associated with the full-potential equation exist in the area of solution-adaptive grids. This would produce improvements in both computational and storage requirements and at the same time be directly applicable to other equation formulations.

Additional basic research associated with the full-potential equation in the areas of computational efficiency or algorithm accuracy or both will still be important in the years to come, but the trend in this regard is seen to be more in the direction of the Euler and Navier-Stokes equations. With these equation sets, more complete and, therefore, more accurate descriptions of the flow field physics, will be possible. The main advantage this provides is the analysis of off-design conditions with significant viscous effects, including massive separation (Navicr-Stokes equations) or in the analysis of flow fields with significant vorticity interaction, for example, in a closely coupled wing-canard configuration (Euler equations). The major problems associated with these equation sets are obtaining sufficient computational efficiency, geometrical generality, and, in the case of the Navier-Stokes equations, adequate turbulence models.

In this regard the full-potential formulation may provide a significant benefit. Formulations that effectively simulate the Euler formulation (for example, the nonisentropic full-potential formulation presented in sec. 6.5) can be developed as a much cheaper alternative to the Euler equations. In another type of application, the standard full-potential formulation could be used to accelerate convergence for the Euler or Navier-Stokes formulations. In many transonic cases of interest, solutions of the full-potential and Euler formulations are very similar. A full-potential solution could be used as an "initial guess" for the Euler calculation, and thus significantly reduce the computational effort associated with the more expensive Euler calculation. In still another application, the full-potential equation could be used in regions of the flow field where the irrotationality and isentropic assumptions are valid, leaving the regions with significant entropy and vorticity gradients for the Euler equations. For many calculations these regions may be very important but confined to only a small portion of the entire flow field. Thus, a large improvement in computational efficiency can be gained by the use of such a hybrid scheme.

In summary, research opportunities associated with the full-potential formulation are numerous and varied. They range from the development of application codes to basic algorithm research. The use of these methods in practical applications to predict transonic flow fields about general three-dimensional configurations is increasing and is expected to help aircraft designers produce more efficient aircraft at less cost.

REFERENCES

1. Hall, M. G.: "Computational Fluid Dynamics — A Revolutionary Force in Aerodynamics." Proceedings of the Fifth AIAA Computational Fluid Dynamics Conference, Palo Alto, Calif., June 1981.

2. Holst, T. L.; Slooff, J. W.; Yoshihara, H.; and Ballhaus, W. F., Jr.: "Applied Computational Transonic Aerodynamics." AGARDograph No. 266, 1982.

3. Chapman, D. R.: "Computational Aerodynamics: Review and Outlook." Dryden Research Lecture presented at the 17th AIAA Aerospace Sciences Meeting, New Orleans, La., Jan. 1979. (See also AIAA J., vol. 17, Dec. 1979, pp. 1293-1313.)

4. Chapman, D. R.: Trends and Pacing Items in Computational Aerodynamics." Presented at the Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford University and NASA Ames Research Center, Moffett Field, Calif., June 1980.

5. Kordulla, W.: "Calculation of 3-D Transonic Flows — Survey of Recent Developments Including Viscous Effects." von Kármán Institute Lecture Series 1980-6, Rhode-Saint-Gènese, Belgium, Mar. 1980.

6. Baker, T. J.: "The Computation of Transonic Potential Flow." von Kármán Institute Lecture Series 1981-5, Rhode-Saint-Gènese, Belgium, Mar. 1981.

7. Lock, R. C.: "A Review of Methods for Predicting Viscous Effects on Aerofoils and Wings at Transonic Speeds." AGARD CP 291, Paper no. 2, 1980.

8. Melnik, R. E.: "Turbulent Interactions on Airfoils at Transonic Speeds — Recent Developments." AGARD CP-291, Paper no. 10, 1980.

9. Ames, W. F.: Numerical Methods for Partial Differential Equations. Academic Press, New York, 1977.

10. Mitchell, A. R.: Computational Methods in Partial Differential Equations. John Wiley and Sons, New York, 1969.

11. Ashley, H.; and Landahl, M.: Aerodynamics of Wings and Bodies. Addison-Wesley Pub. Co., Inc., Reading, Mass., 1965.

12. Steger, J. L.; and Baldwin, B. S.: "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow." NASA TN D-6997, 1972.

13. Lax, P. D.: "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation." Commun. Pure Appl. Math., vol. 7, 1954, pp. 159-193.

14. Newmann, P. A.; and South, J. C., Jr.: "Conservative Versus Nonconservative Differencing: Transonic Streamline Shape Effects." NASA TM X-72827, 1976.

15. O'Brien, C. G.; Hyman, M. A.; and Kaplan, S.: "A Study of the Numerical Solution of Partial Differential Equations." J. Math. Phys., vol. 29, 1951, pp. 223-251.

16. Smith, G. D.: Numerical Solution of Partial Differential Equations: Finite Difference Methods. Second ed., Clarendon Press, Oxford, 1978.

17. Richtmyer, R. D.; and Morton, K. W.: Difference Methods for Initial Value Problems. Second ed., Interscience Publishers, New York, 1967.

18. Garabedian, P. R.: "Estimation of the Relaxation Factor for Small Mesh Size." Math. Tab. Natl. Res. Coun., vol. 10, 1956, pp. 183-185.

19. Martin, E. D.; and Lomax, H.: "Rapid Finite-Difference Computation of Subsonic and Slightly Supercritical Aerodynamic Flows." AIAA J., vol. 13, May 1975, pp. 579-586.

20. Martin, E. D.: "A Fast Semidirect Method for Computing Transonic Aerodynamic Flows." Proceedings of the Second AIAA Computational Fluid Dynamics Conference, Hartford, Conn., June 1975, pp. 162-174. (See also AIAA J., vol. 14, July 1976, pp. 914-922; and Errata, AIAA J., vol. 14, Nov. 1976, p. 1664.)

21. Martin, E. D.: "A Split-Recoupled-Semidirect Computational Technique Applied to Transonic Flow over Lifting Airfoils." AIAA Paper 78-11, Jan. 1978.

22. Caspar, J. R.; Hobbs, D. E.; and Davis, R. L.: "The Calculation of Two-Dimensional Compressible Potential Flow in Cascades Using Finite Area Techniques." AIAA Paper 79-77, Jan. 1979.

23. Casper, J. R.: A Model Problem Study of Transonic Potential Flow Procedures." AIAA Paper 80-337, Jan. 1980.

24. Murman, E. M.; and Cole, J. D.: "Calculation of Plane Steady Transonic Flows." AIAA J., vol. 10, Jan. 1971, pp. 114-121.

25. Steger, J. L.; and Lomax, H.: "Numerical Calculation of Transonic Flow About Two-Dimensional Airfoils by Relaxation Procedures." AIAA J., vol. 10, Jan. 1972, pp. 49-54.

26. Garabedian, P. R.; and Korn, D.: "Analysis of Transonic Airfoils." Commun. Pure Appl. Math., vol. 24, 1972, pp. 841-851.

27. Bailey, F. R.; and Steger, J. L.: "Relaxation Techniques for Three-Dimensional Transonic Flow About Wings." AIAA Paper 72-189, Jan. 1972.

28. Ballhaus, W. F., Jr.; and Bailey, F. R.: "Numerical Calculation of Transonic Flow About Swept Wings." AIAA Paper 72-677, June 1972.

29. Jameson, A.: "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1." Commun. Pure Appl. Math., vol. 27, 1974, pp. 283-309.

30. Lapidus, A.: "A Detached Shock Calculation by Second-Order Finite Differences." J. Comp. Phys., vol. 2, 1967, pp. 154-177.

31. Viviand, H.: "Conservative Forms of Gas Dynamic Equations." La Recherche Aerospatiale, no. 1, Jan.-Feb. 1974, pp. 65-68.

32. Vinokur, M.: "Conservative Equations of Gas Dynamics in Curvilinear Coordinate Systems. J. Comp. Phys., vol. 14, Feb. 1974, pp. 105-125.

33. Steger, J. L.: "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils." AIAA Paper 77-665, June 1977. (See also AIAA J., vol. 16, July 1978, pp. 679-686.)

34. Holst, T. L.: "Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh." AIAA Paper 78-1113, July 1978. (See also AIAA J., vol. 17, Oct. 1979, pp. 1038-1045.)

35. Thompson, J. F.; Warsi, U. A.; and Mastin, C. W.: "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations — A Review." J. Comp. Physics, vol. 47, July 1982, pp. 1-108.

36. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies." J. Comp. Phys., vol. 15, 1974, pp. 299-319.

37. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: "TOMCAT — A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies." J. Comp. Phys., vol. 24, 1977, pp. 274-302.

38. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies." NASA CR-2729, 1977.

39. Thomas, P. D.; and Middlecoff, J. F.: "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations." AIAA J., vol. 18, no. 6, June 1980, pp. 652-656.

40. Steger, J. L.; and Sorenson, R. L.: "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations." J. Comp. Phys., vol. 33, no. 3, Dec. 1979, pp. 405-410.

41. Sorenson, R. L.; and Steger, J. L.: "Numerical Generation of Two-Dimensional Grids by the Use of Poisson Equations with Grid Control at Boundaries." Proceedings of the Numerical Grid Generation Techniques Workshop, Langley Research Center, NASA, Hampton, Va., NASA CP-2166, 1980, pp. 449-461.

42. Starius, G.: "Constructing Orthogonal Curvilinear Meshes by Solving Initial Value Problems." Numerische Mathematik, vol. 28, 1977, pp. 25-48.

43. Steger, J. L.; and Chaussee, D. S.: "Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations." Flow Simulations, Inc. Report 80-1, Jan. 1980.

44. Nakamura, S.: "Marching Grid Generation Using Parabolic Partial Differential Equations." Proceedings of the Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Use in the Numerical Solution of Partial Differential Equations, Nashville, Tenn., Apr. 1982.

45. Nakamura, S.: "Noniterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations." Proceedings of the Eighth International Conference on Numerical Methods in Fluid Dynamics, Aachen, W. Germany, June 1982.

46. Sells, C. C. L.: "Plane Subcritical Flow Past a Lifting Aerofoil." Proc. R. Soc. London, vol. 308A, 1968, pp. 377-401.

47. Kacprzynski, J. J.: "Transonic Flow Field Past 2-D Airfoils Between Porous Wind Tunnel Walls with Nonlinear Characteristics." AIAA Paper 75-81, Jan. 1975.

48. Ives, D. C.: "A Modern Look at Conformal Mapping Including Multiple Connected Regions." AIAA J., vol. 14, no. 8, Aug. 1976, pp. 1006-1011.

49. Caughey, D. A.; and Jameson, A.: "Accelerated Iterative Calculation of Transonic Nacelle Flow Fields." AIAA Paper 76-100, Jan. 1976. (See also AIAA J., vol. 15, no. 10, Oct. 1977, pp. 1474-1480.)

50. Chen, L. T.; and Caughey, D. A.: "Calculation of Transonic Inlet Flowfields Using Generalized Coordinates." AIAA Paper 79-0012, Jan. 1979. (See also J. of Aircraft, vol. 17, no. 3, Mar. 1980, pp. 167-174.)

51. Ives, D. C.; and Liutermoza, J. F.: "Analyses of Transonic Cascade Flow Using Conformal Mapping and Relaxation Technqieus." AIAA J., vol. 25, no. 5, May 1977, pp. 647-652.

52. Jameson, A.; and Caughey, D. A.: "A Finite Volume Method for Transonic Potential Flow Calculations." AIAA Paper 77-635, June 1977, pp. 35-54.

53. Caughey, D. A.; and Jameson, A.: "Numerical Calculation of Transonic Potential Flow About Wing-Body Combinations." AIAA J., vol. 17, no. 2, Feb. 1979, pp. 175-181.

54. Caughey, D. A.; and Jameson, A.: "Progress in Finite-Volume Calculations for Wing-Fuselage Combinations." AIAA J., vol. 18, no. 11, Nov. 1980, pp. 1281-1288.

55. Ives, D. C.; and Menor, W. A.: "Grid Generation for Inlet and Inlet-Centerbody Configurations Using Conformal Mapping and Stretching." Proceedings of the Fifth AIAA Computational Fluid Dynamics Conference, Palo Alto, Calif., June 1981, pp. 33-42.

56. Eiseman, P. R.: "A Coordinate System for a Viscous Transonic Cascade Analysis." J. Comp. Phys., vol. 26, no. 3, Mar. 1978, pp. 307-338.

57. Eiseman, P. R.: "A Multi-Surface Method of Coordinate Generation." J. Comp. Phys., vol. 33, no. 1, Oct. 1979, pp. 118-150.

58. Eiseman, P. R.: "Three-Dimensional Coordinates About Wings." Proceedings of the Fourth AIAA Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979, pp. 166-174.

59. Eiseman, P. R.: "Coordinate Generation with Precise Controls." ICASE Report 80-16, June 1980, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Va.

60. Eiseman, P. R.: "Coordinate Generation with Precise Controls over Mesh Properties." ICASE Report 80-30, Nov. 1980, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Va.

61. Pulliam, T. H.; Jespersen, D. C.; and Childs, R. E.: "An Enhanced Version of an Implicit Code for the Euler Equations." AIAA Paper 83-0344, Jan. 1983.

62. McNally, W. D.: "FORTRAN Program for Generating a Two-Dimensional Orthogonal Mesh Between Two Arbitrary Boundaries." NASA TN D-6766, 1972.

63. Graves, R. A., Jr.: "Application of a Numerical Orthogonal Coordinate Generator to Axisymmetric Blunt Bodies." NASA TM-80131, 1979.

64. Eriksson, L. E.: "Three-Dimensional Spline-Generated Coordinate Transformations for Grids around Wing-Body Configurations." Proceedings of the Numerical Grid Generation Techniques Workshop, NASA Langley Research Center, Hampton, Va., NASA CP-2166, 1980, pp. 253-264.

65. Eriksson, L. E.: "Generation of Boundary Conforming Grids Around Wing-Body Configurations Using Transfinite Interpolation." AIAA J., vol. 20, no. 10, Oct. 1982, pp. 1313-1320.

66. Rizzi, A.: "Damped Euler-Equation Method to Compute Transonic Flow Around Wing-Body Combinations." AIAA J., vol. 20, no. 10, Oct. 1982, pp. 1321-1328.

67. Dwyer, H. A.; Kee, R. J.; and Sanders, B. R.: "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer." AIAA J., vol. 18, no. 10, Oct. 1980, pp. 1205-1212.

68. Glowinski, R.: "On Grid Optimization for Boundary Value Problems." Report no. STAN-CS-79-720, Stanford U., Stanford, Calif., Feb. 1979.

69. Pierson, B. L.; and Kutler, P.: "Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics." AIAA J., vol. 18, no. 1, Jan. 1980, pp. 49-54.

70. Rai, M. M.; and Anderson, D. A.: "Grid Evolution in Time Asymptotic Problems." J. of Comp. Phys., vol. 43, no. 2, Oct. 1981, pp. 327-344.

71. Rai, M. M.; and Anderson, D. A.: "Application of Adaptive Grids to Fluid-Flow Problems with Asymptotic Solutions." AIAA J., vol. 20, no. 4, Apr. 1982, pp. 496-502.

72. Rai, M. M.; and Anderson, D. A.: "The Use of Adaptive Grids in Conjunction with Shock Capturing Methods." AIAA Paper 81-1012, June 1981.

73. Ushimaru, K.: "Development and Application of Adaptive Grids in Two-Dimensional Transonic Calculations." AIAA Paper 82-1016, June 1982.

74. Holst, T. L.; and Brown, D.: "Transonic Airfoil Calculations Using Solution Adaptive Grids." AIAA Paper 81-1010, June 1981.

75. Nakamura, S.; and Holst, T. L.: "A New Solution-Adaptive Grid Generation Method for Transonic Airfoil Flow Calculations." NASA TM-81330, 1981.

76. Caughey, D. A.; and Jameson, A.: "Basic Advances in the Finite-Volume Method for Transonic Potential Flow Calculations." Proceedings of the Symposium for the Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, Calif., Jan. 1981.

77. Caughey, D. A.: "A (Limited) Perspective on Computational Aerodynamics." Presented at the AIAA 13th Fluid and Plasma Cynamics Conference, Snowmass, Colo., July 1980.

78. Chen, L. T.: "Higher Accuracy Finite-Difference Schemes for Transonic Airfoil Flowfield Calculations." AIAA Paper 81-0318, Jan. 1981.

79. Chen, L. T.: "Higher-Accuracy Finite-Difference Schemes for Transonic Airfoil Flow Field Calculations." AIAA Paper 81-0381, Jan. 1981.

80. Chen, L. T.: "A More Accurate Transonic Computational Method for Wing-Body Configurations." AIAA Paper 82-0162, Jan. 1982.

81. Deconinck, H.; and Hirsch, C.: "Transonic Flow Calculations with Higher Order Finite Elements." Presented at the 7th International Conference for Numerical Methods in Fluid Dynamics. NASA Ames Research Center, and Stanford U., Moffett Field, Calif., July 1980.

82. Eberle, A.: "A Finite Volume Method for Calculating Transonic Potential Flow Around Wings from the Pressure Minimum Integral." NASA TM-75324, 1978. (Translated from "Eine Methode der Finiten Volumen zur Berechnung der Transsonischen Potentialstromung um Flugel aus dem Druckmininum-integral." MBB-UFE1407(0), Feb. 1978.)

83. Holst, T. L.; and Ballhaus, W. F., Jr.: "Fast Conservative Schemes for the Full Potential Equation Applied to Transonic Flows." NASA TM-78469, 1978. (See also AIAA J., vol. 17, Feb. 1979, pp. 145-152.)

84. Hafez, M. M.; Murman, E. M.; and South, J. C., Jr.: "Artificial Compressibility Methods for Numerical Solution of Transonic Full Potential Equation." AIAA Paper 78-1148, July 1978. (See also AIAA J., vol. 17, Aug. 1979, pp. 838-844.)

85. Jameson, A.: Transonic Potential Flow Calculations Using Conservative Form." Proceedings of the AIAA 2nd Computational Fluid Dynamics Conference, June 1975, pp. 148-155.

86. Holst, T. L.; and Albert, J.: "An Implicit Algorithm for the Conservative, Transonic Full-Potential Equation with Effective Rotated Differencing." NASA TM-78570, 1979.

87. South, J. C., Jr.; Keller, J. D.; and Hafez, M. M.: "Vector Processor Algorithms for Transonic Flow Calculations." Proceedings of the Fourth AIAA Computational Fluid Dynamics Conference, July 1979. (See also AIAA J., vol. 18, no. 7, July 1980, pp. 786-792.)

88. Farrell, C.; and Adamczyk, J.: "Full Potential Solution of Transonic Quasi-3-D Flow Through a Cascade Using Artificial Compressibility." ASME Paper 81-GT-70, Mar. 1981.

89. Akay, H. U.; and Ecer, A.: "Transonic Flow Computations in Cascades Using Finite Element Method," ASME J. Eng. Power, vol. 103, Oct. 1981, pp. 657-664.

90. Deconinck, H.; and Hirsch, C.: "Finite Element Methods for Transonic Blade-to-Blade Calculation in Turbomachines." ASME J. Eng. Power, vol. 103, Oct. 1981, pp. 665-677.

91. Shankar, V.: "Conservative Full Potential, Implicit Marching Scheme for Supersonic Flows." AIAA J., vol. 20, no. 11, Nov. 1982, pp. 1508-1514.

92. Eberle, A.: "Evaluation of a Minimum Principle for Transonic Flow Computations by Finite Elements." Presented at the GAMM Conference on Numerical Methods in Fluid Dynamics, Cologne, W. Germany, 1979.

93. Eberle, A.: "Finite Element Methods for the Solution of Full Potential Equation in Transonic Steady and Unsteady Flow." Presented at the Third International Conference on Finite Elements in Flow Problems, Banff, Alberta, Canada, June 1980.

94. Steger, J. L.; and Caradonna, F. X.: "A Conservative Implicit Finite Difference Algorithm for the Unsteady Transonic Full Potential Equation." AIAA Paper 80-1368, July 1980.

95. Goorjian, P. M.: "Implicit Computations of Unsteady Transonic Flow Governed by the Full Potential Equation in Conservative Form." AIAA Paper 80-0150, Jan. 1980.

96. Habashi, W. G.; and Hafez, M.M.: "Finite Element Solutions of Transonic Flow Problems." AIAA J., vol. 20, no. 10, Oct. 1982, pp. 1368-1376.

97. Godunov, S. K.: Mat. Sb., 47 (1959), p. 271. (See also Cornell Aeronautical Lab. (Calspan) Translation.)

98. Engquist, B.; and Osher, S.: "Stable and Entropy Satisfying Approximations for Transonic Flow Calculations." Math. Comp., vol. 34, 1980, pp. 45-75.

99. Goorjian, P. M.; and Van Buskirk, R.: "Implicit Calculations of Transonic Flows Using Monotone Methods." AIAA Paper 81-0331, Jan. 1981.

100. Goorjian, P. M.; Meagher, M. C.; and Van Buskirk, R.: "Monotone Implicit Algorithms for the Small-Disturbance and Full Potential Equations Applied to Transonic Flows." AIAA Paper 83-0371, Jan. 1983.

101. Boerstoel, J. W.: "A Multigrid Algorithm for Steady Transonic Potential Flows Around Aerofoils Using Newton Iteration." NASA CP-2202, 1981.

102. Slooff, J. W.: "Some New Developments in Exact Integral Equation Formulations for Sub- or Transonic Compressible Potential Flow." Presented at the International School of Applied Aerodynamics, First Course on Computational Methods in Potential Aerodynamics, International Center for Transportation Studies, Amalfi, Italy, May 31 - June 5, 1982.

103. van Leer, B.: "On the Relation Between the Upwind-Differencing Schemes of Godunov, Engquist-Osher and Roe." ICASE Report 81-11, Mar. 1981, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia.

104. Pulliam, T. H.; and Steger, J. L.: "Implicit Finite-Difference Simulation of Three-Dimensional Compressible Flow." AIAA J., vol. 18, Feb. 1980, pp. 159-167.

105. Thomas, P. D.; and Lombard, C. K.: "The Geometric Conservation Law — A Link Between Finite-Difference and Finite Volume Methods of Flow Computation on Moving Grids." AIAA Paper 78-1208, July 1978.

106. Hindman, R. C.: "Geometrically Induced Errors and Their Relationship to the Form of the Governing Equations and the Treatment of Generalized Mappings." AIAA Paper 81-1008, June 1981. (See also AIAA J., vol. 20, no. 10, Oct. 1982, pp. 1359-1367.)

107. Chattot, J. J.; Coulombeix, C.; and da Silva Tome, C.: "Calculs d'ecoulements Transsoniques Autour d'ailes." La Recherche Aerospatiale, no. 4, 1978, pp. 143-159.

108. Chattot, J. J.; and Coulombeix, C.: "Relaxation Method for the Full-Potential Equation." Presented at the GAMM Workshop on Numerical Methods for the Computation of Inviscid Transonic Flow with Shock Waves, Stockholm, Sept. 1979. (See also ONERA Report no. 1979-154.)

109. Flores, J.; Holst, T. L.; Kwak, D.; and Batiste, D. M.: "A New Consistent Spatial Differencing Scheme for the Transonic Full Potential Equation." AIAA Paper 83-0373, Jan. 1983.

110. Viviand, H.: "Conservative Forms of Gas Dynamic Equations." La Recherche Aerospatiale, no. 1, Jan.-Feb. 1974, pp. 65-68.

111. Dougherty, F. C.; Holst, T. L.; Gundy, K. L.; and Thomas, S. D.: "TAIR — A Transonic Airfoil Analysis Computer Code." NASA TM-81296, 1981.

112. Bauer, F.; Garabedian, P.; Korn, D.; and Jameson, A.: Supercritical Wing Sections II. Vol. 108, Springer-Verlag, New York, 1975.

113. Klopfer, G. H.; and Nixon, D.: "Nonisentropic Potential Formulation for Transonic Flows." Presented at the AIAA 21st Aerospace Sciences Meeting, Reno, Nev., Jan. 1983.

114. Steinhoff, J.; and Jameson, A.: "Multiple Solutions of the Transonic Full Potential Equations." AIAA J., vol. 20, no. 11, Nov. 1982, pp. 1521-1525.

115. Piers, W. J.; and Slooff, J. W.: "Calculation of Transonic Flow by Means of a Shock-Capturing Field Panel Method." Proceedings of the AIAA Fourth Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979.

116. Vigneron, Y.; Brocard, O.; Bousquet, J.; and Lejal, T.: "Une Methode Variationelle d'Elements Finis Pour la Resolution d'Ecoulements Transoniques Tridimensionnels." AGARD CP-285, May 1980.

117. Bristeau, M. O.; Glowinski, R.; Periaux, J.; Perrier, P.; Pironneau, O.; and Poirier, G.: "Applications of Optimal Control and Finite Element Methods to the Calculation of Transonic Flows and Incompressible Viscous Flows." Numerical Methods in Applied Fluid Dynamics, B. Hunt, ed., Academic Press, London, 1980.

118. Periaux, J.: "Resolution de Quelques Problems Non Lineaires en Aerodynamique par des Methods d'Elements Finis et de Moindres Carres Fonctionnels." These de 3eme Cycle, Universite Pierre et Marie Curie, Paris, France, June 1979.

119. Bristeau, M. O.; Glowinski, R.; Periaux, J.; Perrier, P.; and Pironneau, O.: "On the Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least Squares and Finite Elements Methods (I) Least Squares Formulations and Conjugate Gradient Solution of the Continuous Problem." Commun. Meth. Appl. Mech. Eng., vol. 17/18, 1979, pp. 619-657.

120. Emmons, H.: "The Numerical Solution of Compressible Fluid Flow Problem." NACA TN-932, 1944.

121. Emmons, H.: "The Theoretical Flow of a Frictionless, Adiabatic, Perfect Gas Inside of a Two-Dimensional Hyperbolic Nozzle." NACA TN-1003, 1946.

122. Emmons, H.: "Flow of a Compressible Fluid Past a Symmetrical Airfoil in a Wind Tunnel and in Free Air." NACA TN-1746, 1948.

123. Hafez, M.; and Lovell, D.: "Numerical Solution of Transonic Stream Function Equation." Proceedings of the AIAA Fifth Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979.

124. Varga, R. S.: Matrix Iterative Analysis. Prentice Hall, Inc., N.J., 1962.

125. Ballhaus, W. F., Jr.; Jameson, A.; and Albert, J.: "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems." AIAA J., vol. 16, no. 6, June 1978, pp. 573-579.

126. Ballhaus, W. F., Jr.; and Steger, J. L.: "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equations." NASA TM X-73082, 1975.

127. South, J. C., Jr.: Private communication, 1982.

128. Melnik, R. E.: "Wake Curvature and Trailing Edge Interaction Effects in Viscous Flow Over Airfoils." Advanced Technology Airfoil Research, NASA CP-2045, Mar. 1978, pp. 255-270.

129. Ballhaus, W. F., Jr.: "A Fast Implicit Solution Procedure for Transonic Flows." Third International Symposium on Computing Methods in Applied Sciences and Engineering, Versailles, France, Dec. 1977.

130. Holst, T. L.: A Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation." Proceedings of the Fourth AIAA Computational Fluid Dynamics Conference, Williamsburg, Va., July 1979, pp. 109-121. (See also AIAA J., vol. 18, No. 12, Dec. 1980, pp. 1431-1439.)

131. Atta, E. H.: "Component-Adaptive Grid Interfacing." AIAA Paper 81-0382, Jan. 1982.

132. Kwak, D.: "An Implicit, Transonic, Full-Potential Code for Cascade Flow on H-Grid Topology." AIAA Paper 83-0506, Jan. 1983.

133. Holst, T. L.; and Thomas, S. D.: "Numerical Solution of Transonic Wing Flow Fields." AIAA Paper 82-0105, Jan. 1982.

134. Atta, E. H.; and Vadyak, J.: "A Grid Interfacing Zonal Algorithm for Three Dimensional Transonic Flows About Aircraft Configurations." AIAA Paper 82-1017, June 1982.

135. Benek, J.; Steinhoff, J.; and Jameson, A.: "Application of Approximate Factorization to Three Dimensional Transonic Potential Flow Calculations." Presented at the Fifth AIAA Computational Fluid Dynamics Conference, Palo Alto, Calif., June 1981.

136. Baker, T. J.: "Potential Flow Calculation by the Approximate Factorization Method." J. Comp. Phys., vol. 42, 1981, pp. 1-19.

137. Baker, T. J.; and Forsey, C. R.: "A Fast Algorithm for the Calculation of Transonic Flow over Wing/Body Combinations." Proceedings of the AIAA Fifth Computational Fluid Dynamics Conference, Palo Alto, Calif., June 1981.

138. Catherall, D.: "Optimum Approximate-Factorisation Schemes for 2D Steady Potential Flows." Proceedings of the AIAA Fifth Computational Fluid Dynamics Conference, Palo Alto, Calif., July 1981, pp. 213-221. (See also AIAA J., vol. 20, no. 8, Aug. 1982, pp. 1057-1063.)

139. Fedorenko, R. P.: "The Speed of Convergence of One Iterative Process." USSR Computational Mathematics and Mathematical Physics, vol. 4, no. 3, 1964, pp. 227-235.

140. Bakhvalov, N. S.: "On the Convergence of a Relaxation Method with Natural Constraints on the Elliptic Operator." USSR Computational Mathematics and Mathematical Physics, vol. 6, no. 5, 1966, pp. 101-135.

141. Nicolaides, R. A.: "On the L2 Convergence of an Algorithm for Solving Finite Element Systems." Math. Comp., vol. 31, 1977, pp. 892-906.

142. Hackbusch, W.: "Convergence of Multi-Grid Iterations Applied to Difference Equations." Report 79-5, Köln University Mathematics Institute, Köln, W. Germany, Apr. 1979.

143. Brandt, A.: "Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems." Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris, France, 1972, Lecture Notes in Physics, vol. 18, pp. 82-89, Springer-Verlag, Berlin and New York, 1973.

144. Brandt, A.: "Multi-Level Adaptive Solutions to Boundary Value Problems." Math. Comp., vol. 31, Apr. 1977, pp. 333-390.

145. South, J. C., Jr.; and Brandt, A.: "Application of Multi-Level Grid Method to Transonic Flow Calculations." ICASE Report no. 76-8, Mar. 1976, Institue for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia.

146. Arlinger, B.: "Multi-Grid Technique Applied to Lifting Transonic Flow Using Full Potential Equation." SAAB Report L-0-1 B439, Dec. 1978.

147. Jameson, A.: "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method." AIAA Paper 79-1458, July 1979.

148. Fuchs, L. J.: "Finite Difference Methods for Plane Steady Inviscid Transonic Flows." TRITA-GAD-2, 1977.

149. Deconinck, H.; and Hirsch, C.: "A Multigrid Method for the Transonic Full Potential Equation Discretized with Finite Elements on an Arbitrary Body Fitted Mesh. NASA CP-2202, 1981, pp. 61-81.

150. Arlinger, B.: "Axisymmetric Transonic Flow Computations Using a Multigrid Method." Presented at the Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford U. and NASA Ames Research Center, Moffett Field, Calif., June 1980.

151. McCarthy, D. R.; and Reyhner, T. A.: "A Multi-Grid Code for Three-Dimensional Transonic Potential Flow about Axisymmetric Inlets at Angle of Attack." AIAA Paper 80-1365, July 1980. (See also AIAA J., vol. 20, no. 1, Jan. 1982, pp. 45-50.)

152. Brown, J. J.: "A Multigrid Mesh-Embedding Technqiue for Three-Dimensional Transonic Potential Flow Analysis." NASA CP-2202, 1981, pp. 131-149.

153. Shmilovich, A.; and Caughey, D. A.: "Application of the Multi-Grid Method to Calculation of Transonic Potential Flow about Wing/Fuselage Combinations." NASA CP-2202, 1981, pp. 101-130.

154. Caughey, D. A.: "Multi-Grid Calculation of Three-Dimensional Transonic Potential Flows." AIAA Paper 83-374, 1983.

155. Stone, H. L.: "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations." SIAM J. Numer. Analysis, vol. 5, no. 3, 1969.

156. Sankar, N. L.; and Tassa, Y.: "An Algorithm for Unsteady Transonic Potential Flow Past Airfoils." Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford U. and NASA Ames Research Center, Moffett Field, Calif., June 1980.

157. Sankar, N. L.: Malone, J.; and Tassa, Y.: "A Strongly Implicit Procedure for Steady Three-Dimensional Transonic Potential Flows." AIAA Paper 81-0385, Jan. 1981.

158. Roach, R. L.; and Sankar, N. L.: "The Strongly Implicit Procedure Applied to the Flow Field of Transonic Turbine Cascades." AIAA Paper 81-0211, Jan. 1981.

159. Sankar, N. L.; Malone, J. B.; and Tassa, Y.: "An Implicit Conservative Algorithm for Steady and Unsteady Three-Dimensional Transonic Potential Flows." Proceedings of the Fifth Computational Fluid Dynamics Conference, Palo Alto, Calif., June 1981.

160. Keller, J. D.; and Jameson, A.: "Preliminary Study of the Use of the STAR-100 Computer for Transonic Flow Calculations." AIAA Paper 78-12, Jan. 1978.

161. Redhead, D. D.; Chen, A. W.; and Hotovy, S. G.: "New Approach to the 3-D Transonic Flow Analysis Using the STAR-100 Computer." AIAA J., vol. 17, no. 1, Jan. 1979, pp. 98-99.

162. Hotovy, S. G.; and Dickson, L. J.: "Evaluation of a Vectorizable 2-D Transonic Finite Difference Algorithm." AIAA Paper 79-276, Jan. 1979.

163. Bailey, F. R.: "A View Toward Future Fluid Dynamics Computing." AIAA Paper 78-1112, July 1978.

164. Holst, T. L.: "An Implicit Algorithm for Solving the Transonic, Conservative Full-Potential Equation." Proceedings of the 1980 Army Numerical Analysis and Computers Conference, ARO Report 80-3, Aug. 1980, pp. 197-222.

165. Hafez, M. M.; and Cheng, H. K.: "Convergence Acceleration of Relaxation Solution for Transonic Flow Computations." AIAA J., vol. 15, Mar. 1977, pp. 329-336.

166. Yu, N. J.; and Rubbert, P. E.: "Acceleration Schemes for Transonic Potential Flow Calculations." AIAA Paper 80-0338, Jan. 1980.

167. Glowinski, R.; Periaux, J.; and Pironneau, O.: "Transonic Flow Simulation by the Finite Element Method Via Optimal Control." Third International Symposium on Finite Elements in Flow Problems, Canada, 1980.

168. Chattot, J. J.; and Coulombeix, C.: "Finite Difference and Finite Element Solutions of the Full Potential Equation in Two Dimensions." Presented at the Fourth GAMM Conference on Numerical Methods in Fluid Mechanics, Paris, France, Oct. 1981.

169. Wong, Y. S.; and Hafez, M. M.: "Application of Conjugate Gradient Methods to Transonic Finite Difference and Finite Element Calculations." Proceedings of the Fifth AIAA Computational Fluid Dynamics Conference, Palo Alto, Calif., July 1981, pp. 272-283. (See also AIAA J., vol. 20, no. 11, Nov. 1982, pp. 1526-1533.)

170. Wong, Y. S.; and Hafez, M. M.: "Preconditioned Conjugate Gradient Methods for Transonic Flow Calculations." ICASE Report no. 81-30, Sept. 1981, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia.

171. Wong, Y. S.; and Hafez, M. M.: "A Minimal Residual Method for Transonic Potential Flow." ICASE Report no. 82-15, June 1982, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virgina.

172. Hafez, M. M.; and Lovell, D.: "Improved Relaxation Schemes for Transonic Potential Calculations." AIAA Paper 83-0372, Jan. 1983.

173. Jameson, A.; Caughey, D. A.; Newman, P. A.; and Davis, R. M.: "A Brief Description of the Jameson-Caughey NYU Transonic Swept-Wing Computer Program FLO22." NASA TM X-73996, 1976.

174. Newman, P. A.; Carter, J. E.; and Davis, R. M.: "Interaction of a Two-Dimensional Strip Boundary Layer with a Three-Dimensional Transonic Swept-Wing Code." NASA TM-78640, 1978.

175. Reyhner, T. A.: "Transonic Potential Flow Around Axisymmetric Inlets and Bodies at Angle of Attack." AIAA J., vol. 15, 1977, pp. 1299-1306.

176. Mercer, J. E.; and Murman, E. M.: "Application of Transonic Potential Calculations to Aircraft and Wind Tunnel Configurations." AGARD CP-285, May 1980.

177. Heckmann, G.: "Etude par la Methode des Elements Finis des Interactions Voilure-Fuselage-Nacelle d'un Avion du Type Falcon a Mach = 0.79." AGARD CP-285, May 1980.

178. Forsey, C. R.; and Carr, M. P.: "The Calculation of Transonic Flow over Three-Dimensional Swept Wings Using the Exact Potential Equation." ARA Memo no. 207, 1978, Aircraft Research Association Limited, Bedford, England. (See also, Proceedings of the DGLR/GARTEur 6 Symposium, Transonic Configurations, Bad Harzburg, W. Germany, June 1978.)

179. Eberle, A.: "Transonic Potential Flow Computations by Finite Elements: Airfoil and Wing Analysis, Airfoil Optimization." Lecture held at the DGLR/GARTEur 6 Symposium, Transonic Configurations, Bad Harzburg, W. Germany, June 1978.

180. Chen, L. T.; and Caughey, D. A.: "Higher-Order, Finite-Difference Scheme for Three-Dimensional Transonic Flow Fields About Axisymmetric Bodies." J. of Aircraft, vol. 17, no. 9, Sept. 1980, pp. 668-676.

181. Yu, N. J.: "Grid Generation and Transonic Flow Calculations for Three-Dimensional Configurations." AIAA Paper 80-1391, July 1980.

182. Yu, N. J.: "Transonic Flow Simulations for Complex Configurations with Surface Fitted Grids." AIAA Paper 81-1258, June 1981.

183. Thomas, S. D.; and Holst, T. L.: "Numerical Computation of Transonic Flow About Wing-Fuselage Configurations on a Vector Computer." AIAA Paper 83-0499, Jan. 1983.

184. Bridgmen, J. O.; Steger, J. L.; and Caradona, F. X.: "A Conservative Finite Difference Algorithm for the Unsteady Transonic Potential Equation in Generalized Coordinates." AIAA Paper 82-1388, Aug. 1982.

185. Hinson, B. L.; and Burdges, K. P.: "An Evaluation of Three-Dimensional Transonic Codes Using New Correlation-Tailored Test Data." AIAA Paper 80-0003, Jan. 1980.

186. Bailey, F. R.; and Ballhaus, W. J., Jr.: "Comparisons of Computed and Experimental Pressures for Transonic Flows About Isolated Wings and Wing-Fuselage Configurations." NASA SP-347, 1975.

187. Boppe, C. W.: "Calculation of Transonic Wing Flows by Grid Embedding." AIAA Paper 77-207, Jan. 1977.

188. Brown, J. J.: "A Multigrid Mesh-Embedding Technique for Three-Dimensional Transonic Potential Flow Analysis." AIAA Paper 82-0107, Jan. 1982.

189. Henne, P. A.; and Hicks, R. M.: "Wing Analysis Using a Transonic Potential Flow Computational Method." NASA TM-78464, 1978.

190. Verhoff, A.; and O'Neil, P. J.: "Extension of FLO Codes to Transonic Flow Prediction for Fighter Configurations." Presented at Transonic Perspective — A Critique of Transonic Flow Research, Workshop held at NASA Ames Research Center, Moffett Field, Calif., Feb. 1981. (See also MCAIR Report No. 81-004, 1981.)

191. Subramanian, N. R.; Holst, T. L.; and Thomas, S. D.: "Recent Applications of the Transonic Wing Analysis Computer Code TWING." NASA TM-84283, 1982.

192. Schmitt, V.; and Charpin, F.: "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers." AGARD Report AR-138, May 1979.

193. Lock, R. C.: "The Prediction of Viscous Effects on Aerofoils in Transonic Flow." Proceedings of the DGLR/GARTEur Symposium, Transonic Configurations, Bad Harzburg, W. Germany, June 1978.

194. "Experimental Data Base for Computer Program Assessment." AGARD-AR-138, 1979.

195. Caughey, D. A.; Newman, P. A.; and Jameson, A.: "Recent Experiences with Three-Dimensional Transonic Potential Flow Calculations." NASA TM-78733, 1978.

196. Bartlett, D. W.: "Wind Tunnel Investigations of Several High Aspect-Ratio Supercritical Wing Configurations on a Wide-Body Fuselage." NASA TM X-71996, 1977.

197. Van der Vooren, J.: "XFL022-NLR, A Modified Version of FL022; Computer Code for the Calculation of Transonic Flow about a Wing, Including Body Influence." Unpublished NLR (National Aerospace Laboratory, Amsterdam, The Netherlands) Memorandum, 1980.

198. Hinson, B. L.; and Burdges, K. P.: "Acquisition and Application of Transonic Wing and Far-Field Test Data for Three-Dimensional Computational Method Evaluation." AFSOR-TR-80-0421, Mar. 1980, Air Force Office of Scientific Research, Bolling Air Force Base, D.C.

199. Streett, C. L.: "Viscous-Inviscid Interaction for Transonic Wing-Body Configurations Including Wake Effects." AIAA Paper 81-1266, June 1981. (See also, AIAA J., vol. 20, no. 7, July 1982, pp. 915-923.)

200. Stock, H. W.: "Integral Method for the Calculation of Three-Dimensional, Laminar and Turbulent Boundary Layers." NASA TM-75320, 1978.

201. Smith, P. D.: "An Integral Prediction Method for Three-Dimensional Compressible Turbulent Boundary Layers." Royal Aeronautical Establishment R&M 3739, 1974.

TABLE 1. CONVERGENCE RATE ESTIMATES FOR VARIOUS RELAXATION SCHEMES (FROM REF. 9)

Algorithm	Number of iterations[a]
Point-Jacobi	$2/\Delta^2$
Point-Gauss-Seidel	$1/\Delta^2$
SOR	$1/2\Delta$
Line-Jacobi	$1/\Delta^2$
Line-Gauss-Seidel	$1/2\Delta^2$
SLOR	$1/2\sqrt{2}\Delta$
ADI	$-\log(\Delta/2)/1.55$

[a] Number of iterations required for a one-order-of-magnitude reduction in error [Num, see eq. (3.45)].

TABLE 2. SOME EXISTING THREE-DIMENSIONAL FULL-POTENTIAL CODES

Code name (ref.)	Date	Eq. type[a]	Iteration scheme[b]	Grid	Remarks (ref.)
FL022 (29,173)	1974	NC	SLOR	Sheared conformal	Isolated wings, viscous option (174)
Reyhner (175,151)	1976	NC	SLOR or MG/SLOR	Sheared Cartesian, nonaligned	Axisymmetric inlets ($\alpha > 0$) with or without center bodies
FL027,FL028, FL030 (52-54)	1977	C	SLOR	Sheared conformal	Wing and wing/body [tunnel walls (176)] [AF version (135)]
Dassault (177)	1977	C	Optimal control conjugate gradient	Finite element	Wing, wing/body and wing/body/nacelle
ONERA (107)	1978	C or NC	SLOR or AF	Sheared conformal	Isolated wings
ARA (178)	1978	NC	SLOR	Sheared conformal	Wing and wing/body [AF version (137)]
Eberle (179)	1978	C	SLOR	Finite element	Isolated wings
Chen and Caughey (180)	1979	C or NC	SLOR	Sheared conformal	Axisymmetric inlets ($\alpha > 0$) with and without centerbodies
Yu (181,182)	1980	C	SLOR	Elliptic solver	Wing/body and wing/body/nacelle
Sankar, Malone, Tassa (157)	1981	C	AF	Sheared conformal	Isolated wings [Unsteady option (159)]
TWING (133,183)	1982	C	AF	Elliptic solver	Wing and wing/body
TUNA (184)	1982	C	AF	Elliptic solver	Isolated wings (Unsteady option)

[a] NC = nonconservative; C = conservative.

[b] SLOR = successive-line over relaxation; MG = multigrid; AF = approximate factorization.

TABLE 3. GRID FEATURES AND COMPUTING TIMES FOR SEVERAL TRANSONIC WING COMPUTER CODES (FROM REF. 185)

Code	Wing chordwise grid points	Wing spanwise grid points	Total field points	Approximate CPU time,[a] min	Number of iterations
Bailey-Ballhaus	37	25	41,000	6	200
FL022	61 (root) 40 (tip)	21	159,000	15	50-100
FL027	51	21	90,000	30	200

[a] CDC 7600 computer.

TABLE 4. A COMPARISON OF CODE EXECUTION SPEED FOR THE ONERA M6 WING AT $M_\infty = 0.84$, $\alpha = 3.06°$ (FROM REF. 183)

	FL028	TWING: scalar (ref. 133)	TWING: vector (ref. 183)	TWING: vector (ref. 183)
Computer	CDC 7600	CDC 7600	CDC 7600	Cray-1S
Field points	53,760	40,050	40,050	40,050
Surface points	1,350	1,513	1,513	1,513
Time for 98% lift, sec	742	64	53	4.8
Ratio	155	13	11	1

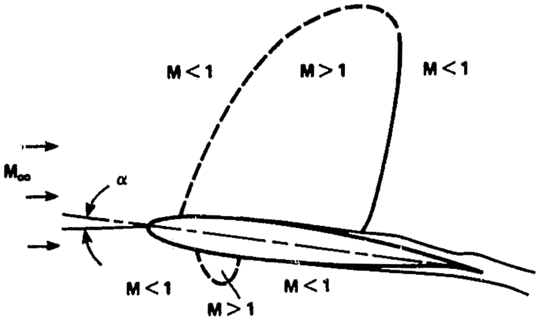


Fig. 1. Typical transonic flow field about an airfoil.

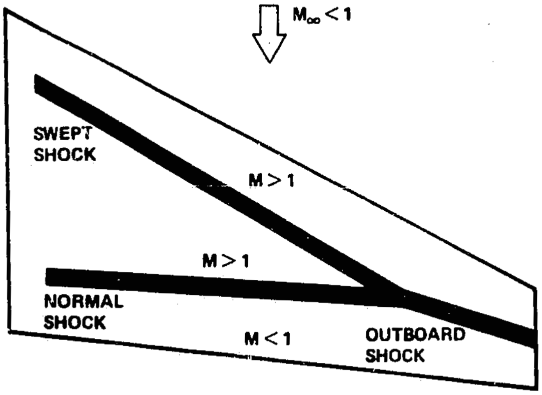


Fig. 2. Typical transonic flow field about a swept wing.

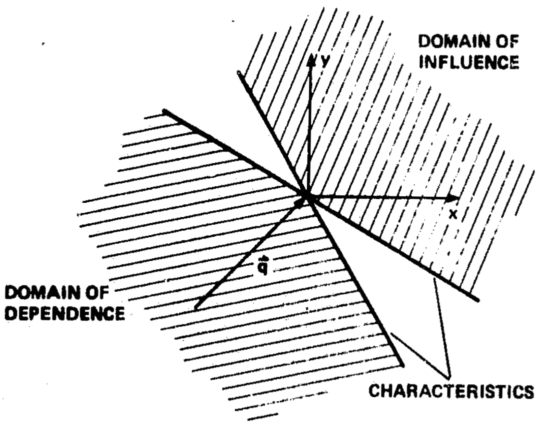


Fig. 3. The domains of dependence and influence for steady supersonic flow.

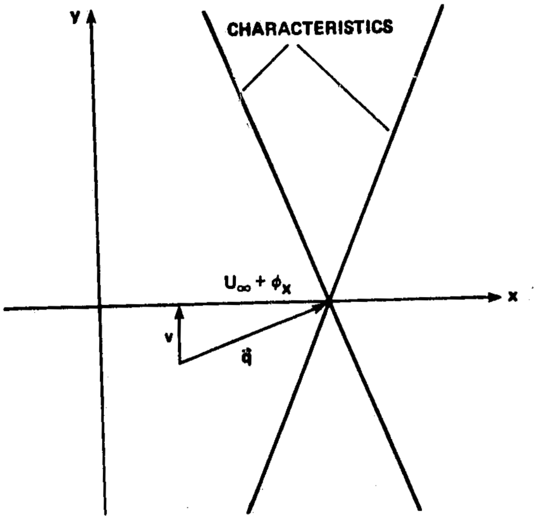


Fig. 4. Characteristics and the TSD equation.

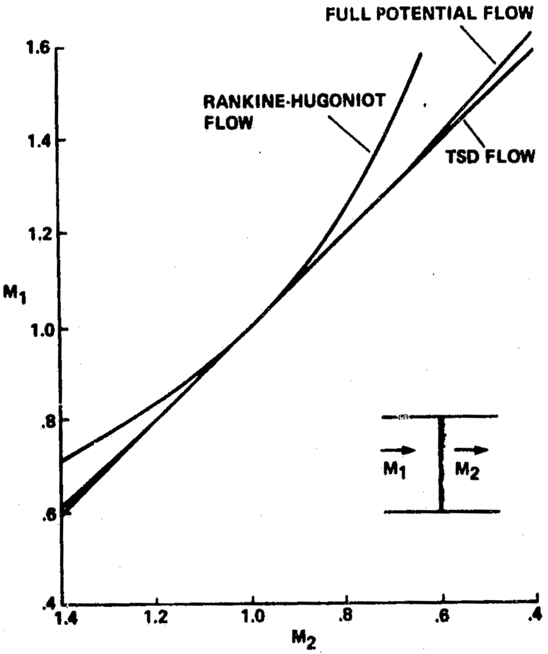


Fig. 5. A comparison of the isentripic shock-jump relation and the Euler shock-jump relations (ref. 12).

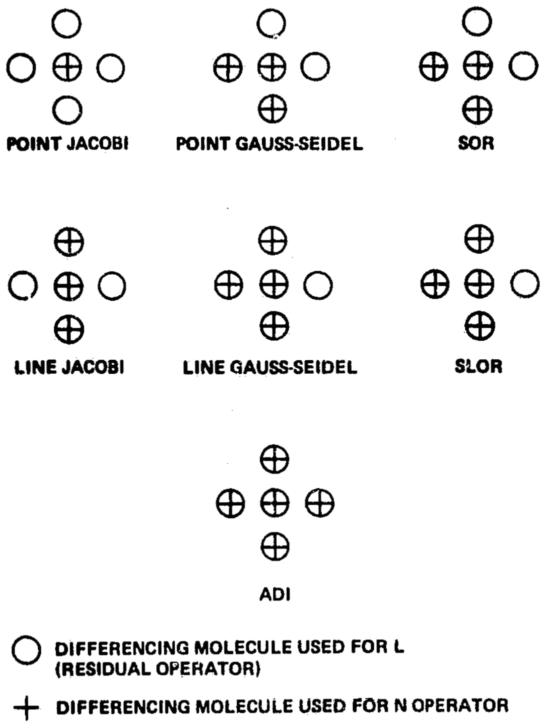

POINT JACOBI
POINT GAUSS-SEIDEL
SOR
LINE JACOBI
LINE GAUSS-SEIDEL
SLOR
ADI
○ DIFFERENCING MOLECULE USED FOR L (RESIDUAL OPERATOR)
+ DIFFERENCING MOLECULE USED FOR N OPERATOR

Fig. 6. A summary of various classical relaxation schemes.

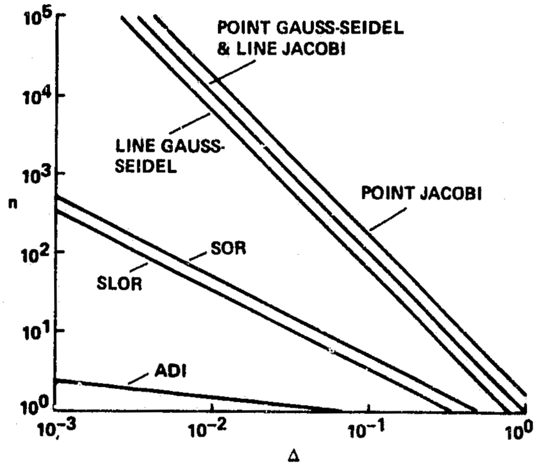


Fig. 7. A comparison of convergence rate estimates for the relaxation schemes of figure 6.

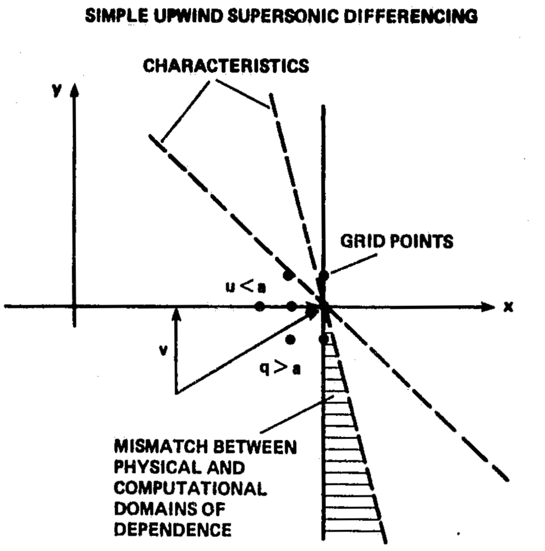

SIMPLE UPWIND SUPERSONIC DIFFERENCING
CHARACTERISTICS
y
x
GRID POINTS
u<a
q>a
v
MISMATCH BETWEEN PHYSICAL AND COMPUTATIONAL DOMAINS OF DEPENDENCE

Fig. 8. Mismatch between the physical and numerical domains of dependence — a destabilizing situation.

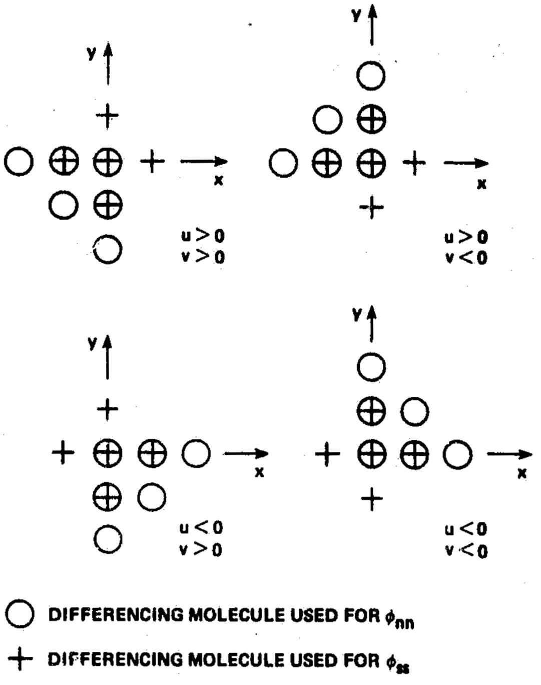
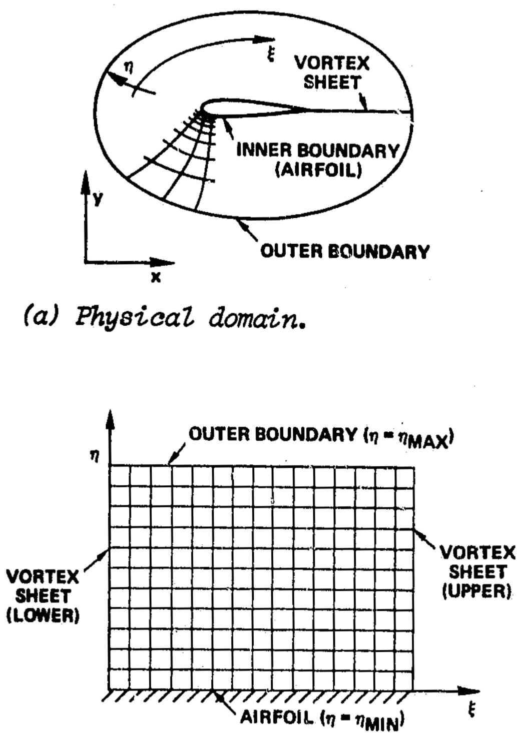

y
x
u>0
v>0
u>0
v<0
u<0
v>0
u<0
v<0
○ DIFFERENCING MOLECULE USED FOR ϕ_{nn}
+ DIFFERENCING MOLECULE USED FOR ϕ_{ss}

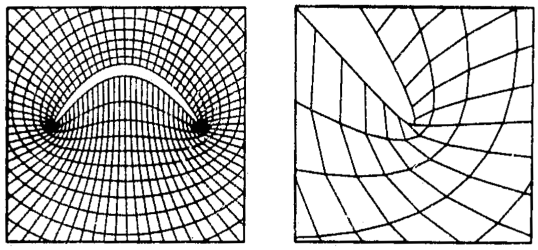
Fig. 9. A schematic representation of the rotated differencing scheme of Jameson (ref. 29).



(a) Physical domain.

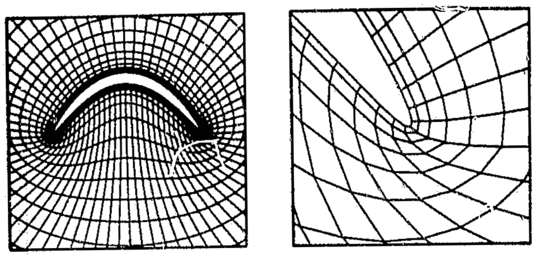
(b) Computational domain.

Fig. 10. Numerically generated airfoil transformation: $(x,y) \rightarrow (\xi,\eta)$, "O" mesh topology.



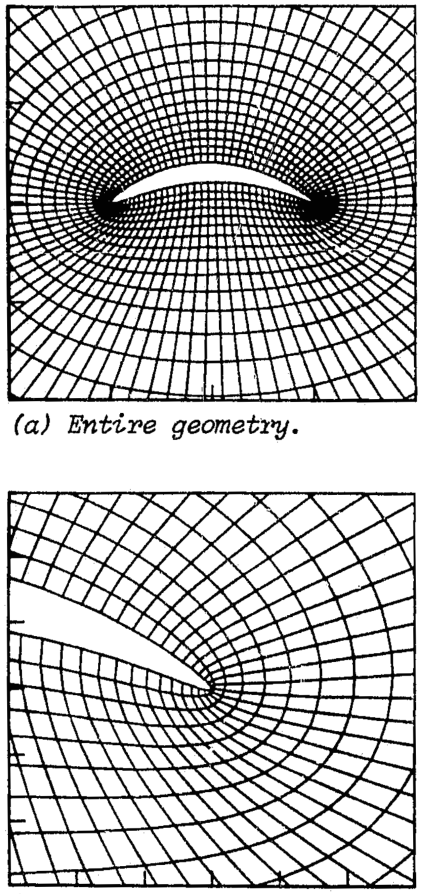
(a) *(b)*

Fig. 11. Finite-difference grid about a highly cambered 12 to 1 ellipse, no control (ref. 41). (a) Entire geometry. (b) Trailing-edge close-up.



(a) *(b)*

Fig. 12. Finite-difference grid about a highly cambered 12 to 1 ellipse, with control terms activated (ref. 41). (a) Entire geometry. (b) Trailing-edge close-up.



(a) Entire geometry.

(b) Close-up of the trailing edge.

Fig. 13. Finite-difference grid about a highly cambered thin ellipse generated via the parabolic scheme of Nakamura (ref. 44).

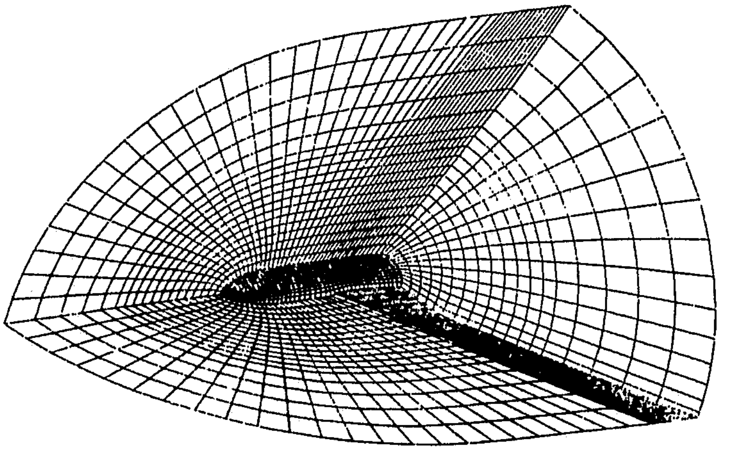
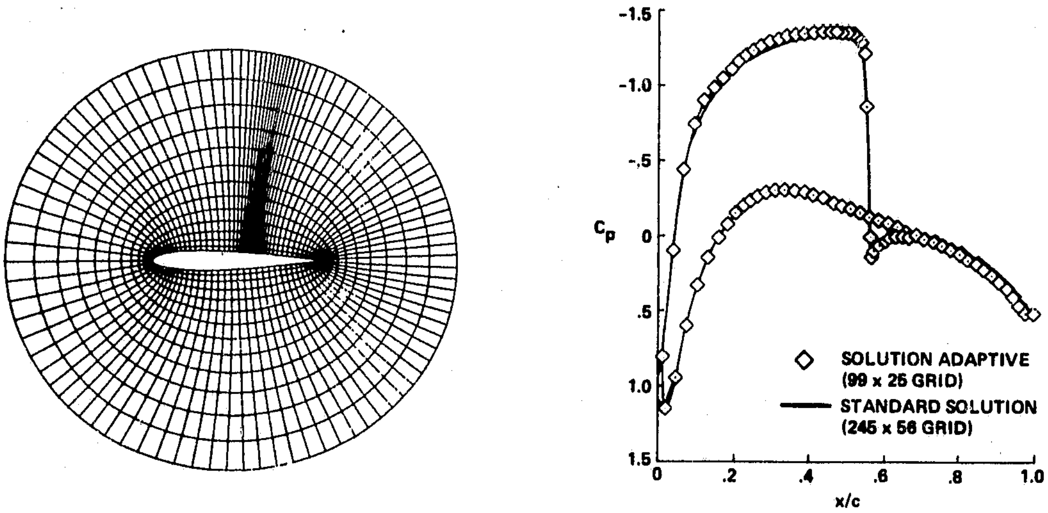


Fig. 14. A wing/fuselage grid generated via the parabolic scheme of Nakamura (ref. 45).



(a) Solution-adaptive grid. *(b) C_p distribution.*

Fig. 15. An example solution-adaptive grid applied to a transonic airfoil calculation: NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 2°$ (ref. 75).

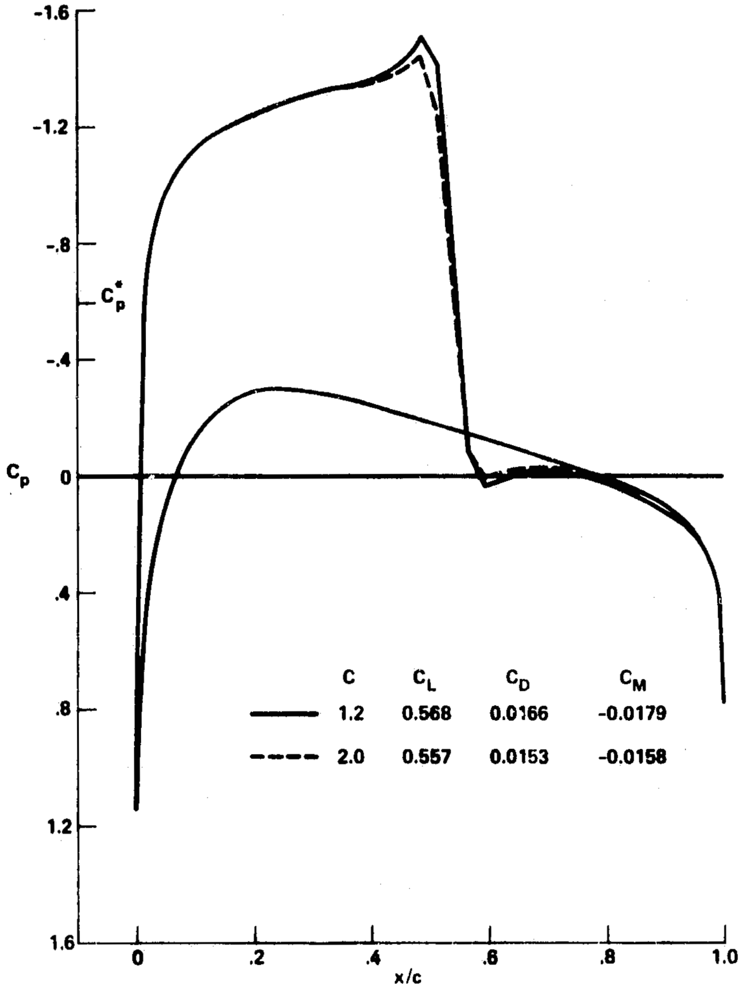


Fig. 16. Pressure coefficient distributions for the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2°$; density computed at node points and artificial viscosity parameter ν computed at half points (ref. 86).

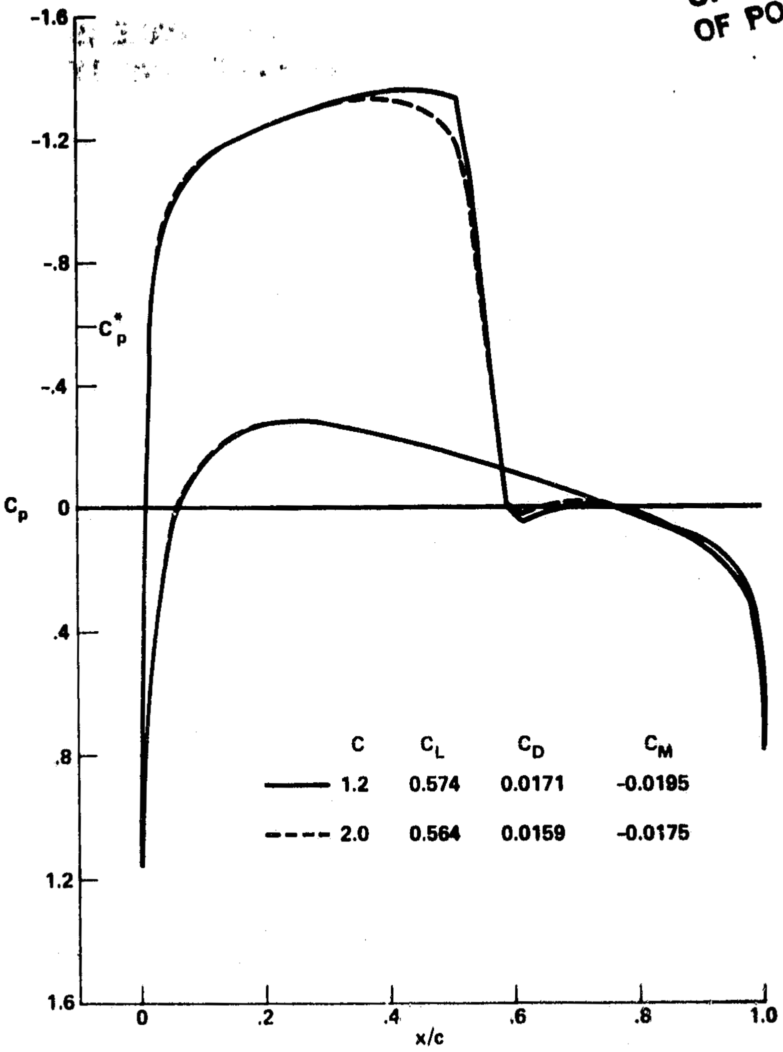


Fig. 17. Pressure coefficient distributions for the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2°$, both density and artificial viscosity parameter ν computed at node points (ref. 86).

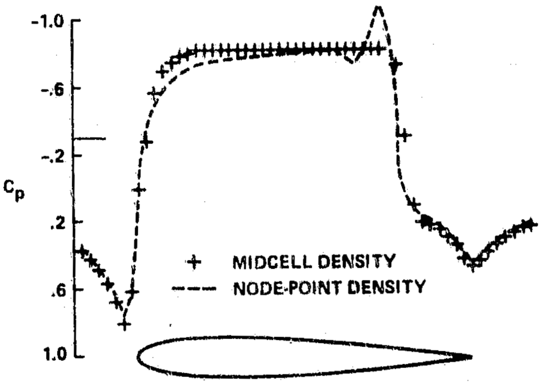
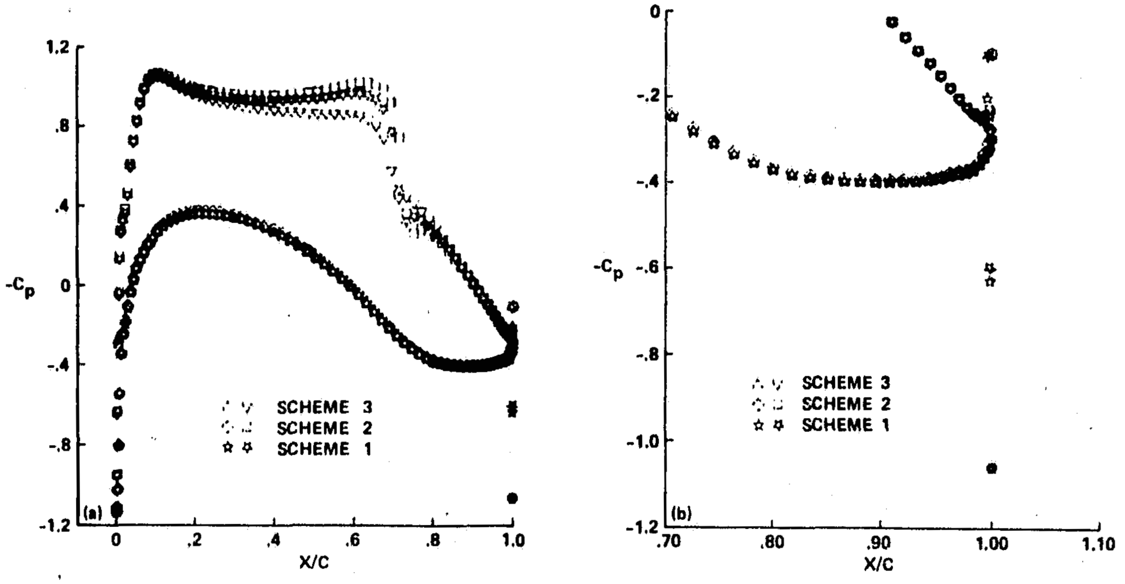


Fig. 18. Pressure coefficient distributions for the NACA 0012 airfoil at $M_\infty = 0.85$ and $\alpha = 0°$, artificial viscosity parameter ν computed at node points (ref. 87).



(a) Entire airfoil. *(b) Close-up of trailing-edge solution.*

Fig. 19. Pressure coefficient distributions for the Korn airfoil at its design point ($M_\infty = 0.75$ and $\alpha = 0.12°$) showing the effects of different spatial differencing schemes on the solution (ref. 109).

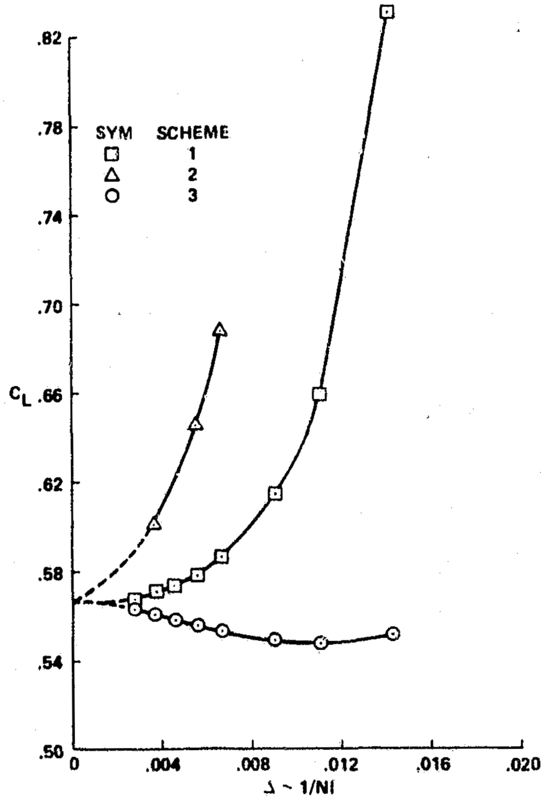


Fig. 20. Mesh refinement study for the NACA 0012 airfoil at $M_\infty = 0.75$, $\alpha = 2°$, showing the effects of grid coarseness on several spatial differencing schemes (ref. 109).

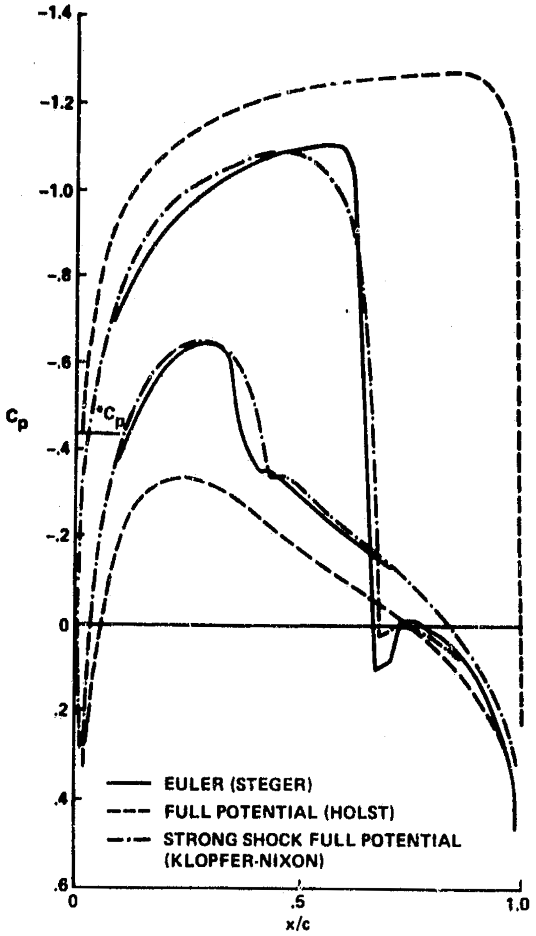


Fig. 21. Airfoil pressure coefficient comparisons of the nonisentropic and isentropic full-potential formulations and the Euler formulation (ref. 113).

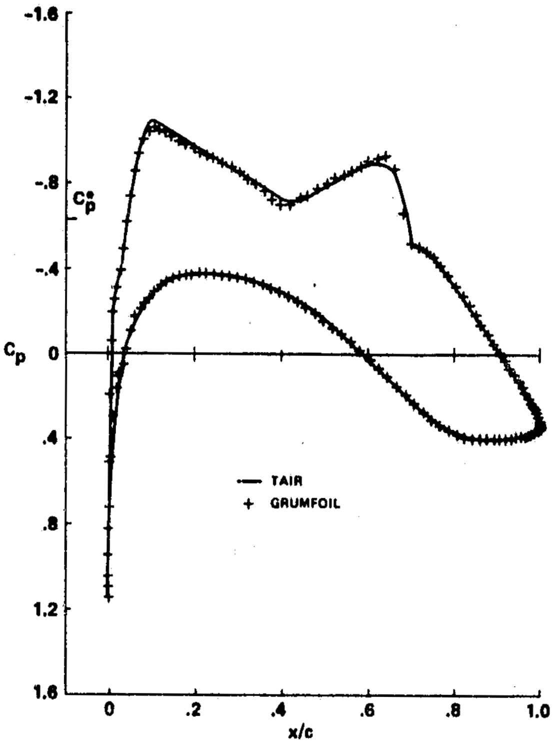


Fig. 22. Pressure coefficient comparison for the Korn airfoil at $M_\infty = 0.74$ and $\alpha = 0°$ (ref. 111).

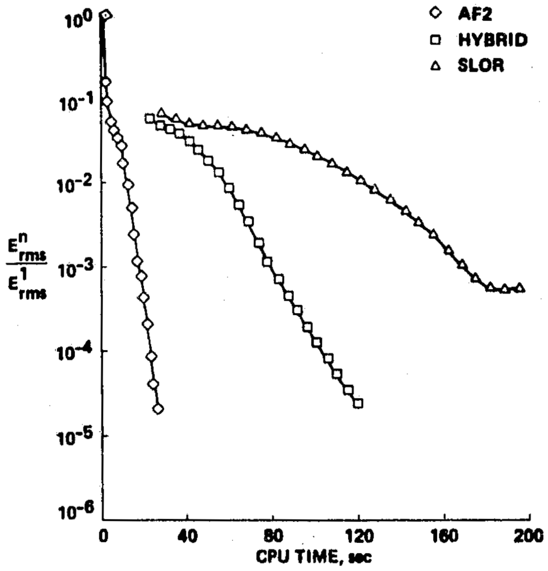


Fig. 23. Convergence history comparison for the Korn airfoil at $M_\infty = 0.74$ and $\alpha = 0°$ (ref. 111).

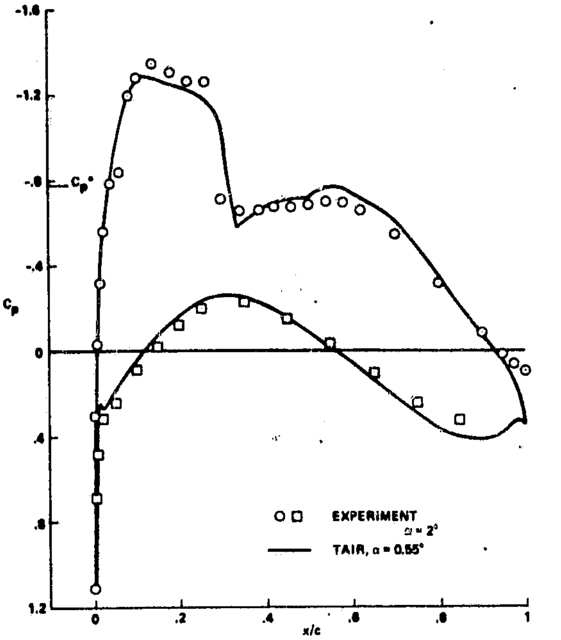


Fig. 24. Pressure coefficient comparison for the CAST 7 supercritical airfoil at $M_\infty = 0.7$ (ref. 111).

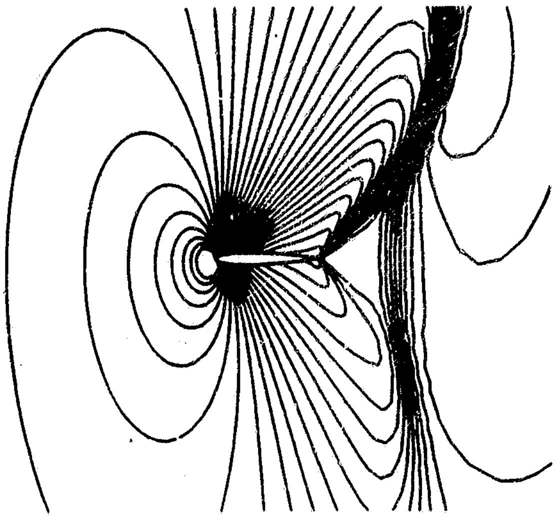


Fig. 25. Mach-number contours about an NACA 0012 airfoil at $M_\infty = 0.95$ and $\alpha = 4°$ (ref. 130).

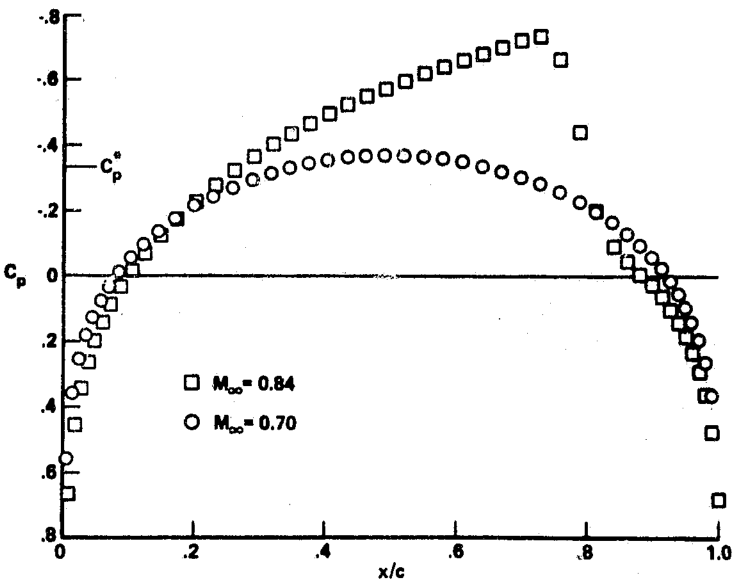


Fig. 26. Subcritical and supercritical solutions used for the convergence history comparisons of figures 27-31 (ref. 83).

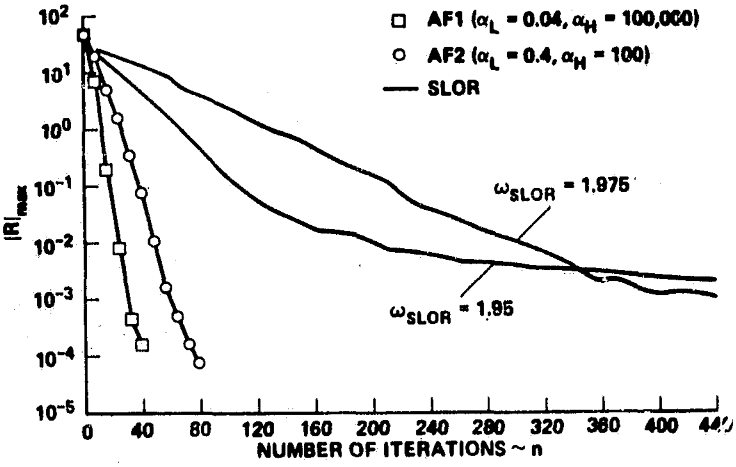


Fig. 27. Maximum residual convergence history comparison, subcritical case, M_∞ = 0.7 (ref. 83).

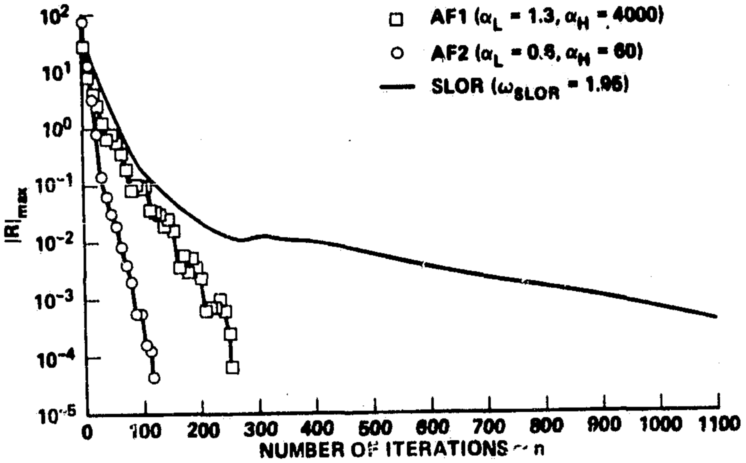


Fig. 28. Maximum residual convergence history comparison, supercritical case, M_∞ = 0.84 (ref. 83).

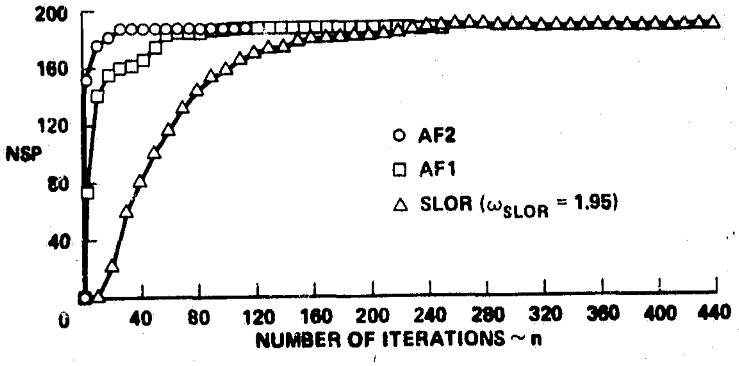
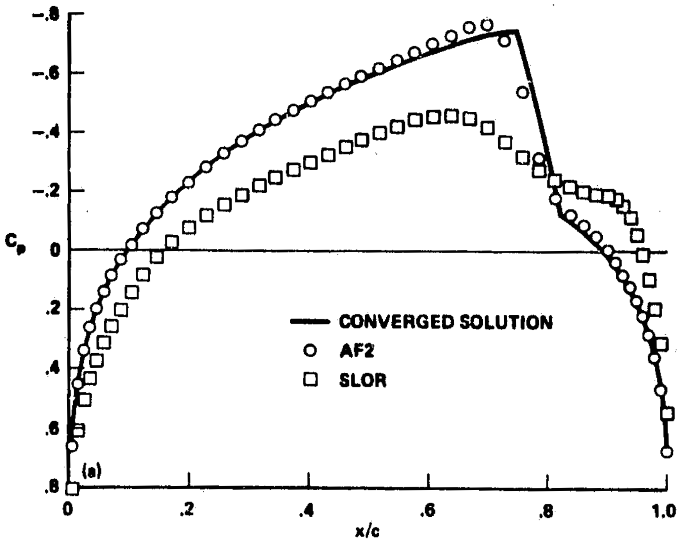
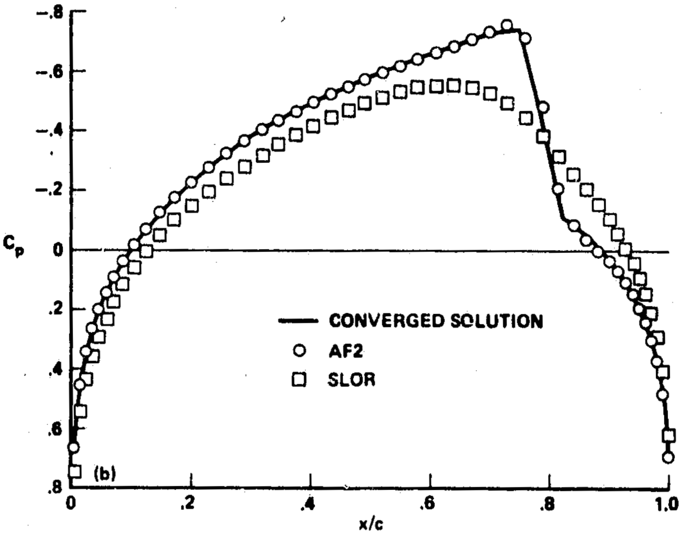


Fig. 29. Development of the supersonic region, M_∞ = 0.84 (ref. 83).

C-2

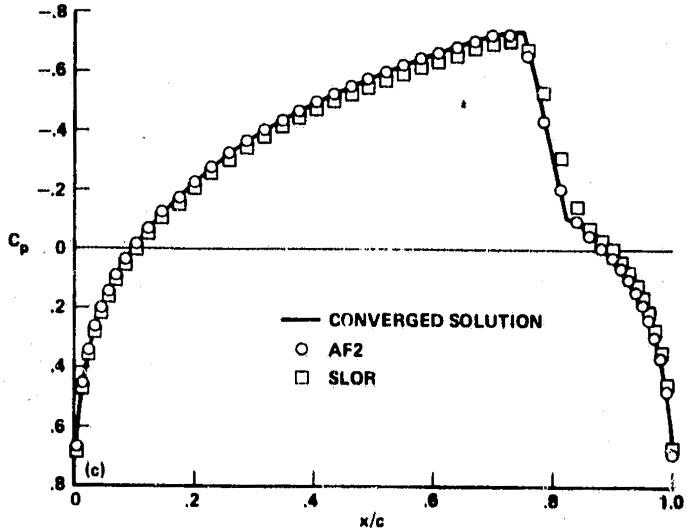


(a) Maximum residual reduced one order of magnitude.



(b) Maximum residual reduced two orders of magnitude.

Fig. 30. Intermediate solution comparisons after specified reductions in the maximum residual, supercritical case, $M_\infty = 0.84$ (ref. 83).



(c) Maximum residual reduced three orders of magnitude.

Fig. 30. Concluded.

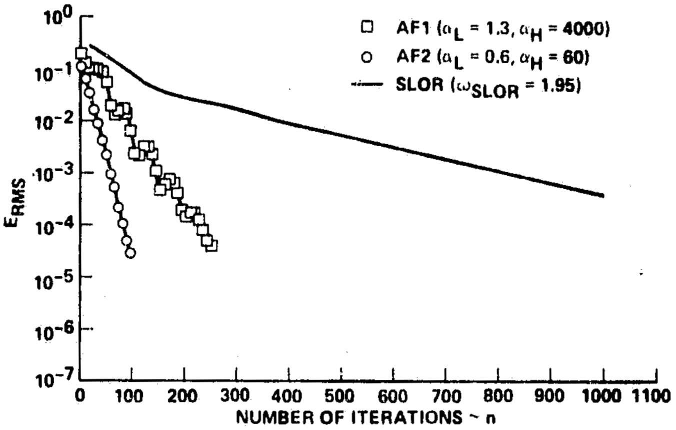


Fig. 31. Root-mean-square error convergence history comparison, supercritical case, M_∞ = 0.84 (ref. 83).

ORIGINAL PAGE IS
OF POOR QUALITY

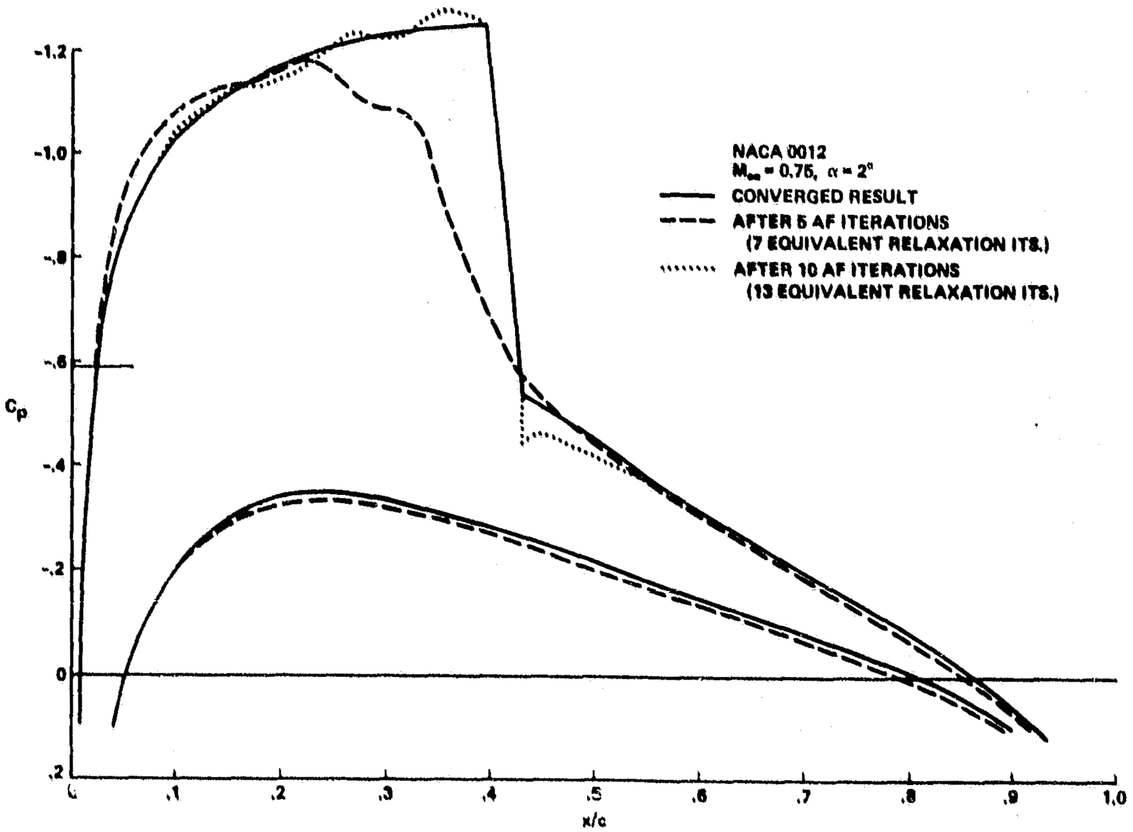


Fig. 32. Development of the pressure coefficient distribution with iteration number for the AF3 iteration scheme, NACA 0012 airfoil, M_∞ = 0.75, α = 2° (ref. 6).

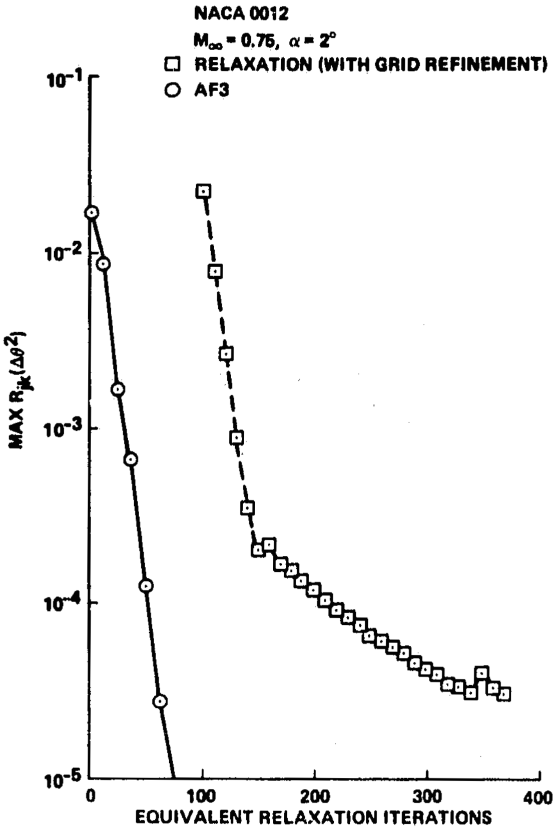


Fig. 33. Convergence history comparison showing the AF3 and SLOR iteration schemes, NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 2°$ (ref. 6).

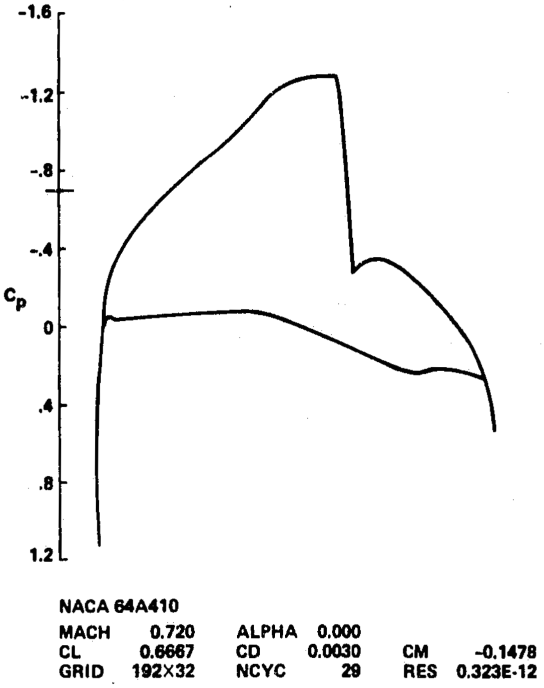


Fig. 34. Converged pressure coefficient distribution from the FL036 multigrid code, NACA 64A010 airfoil, $M_\infty = 0.72$, $\alpha = 0°$ (ref. 147).

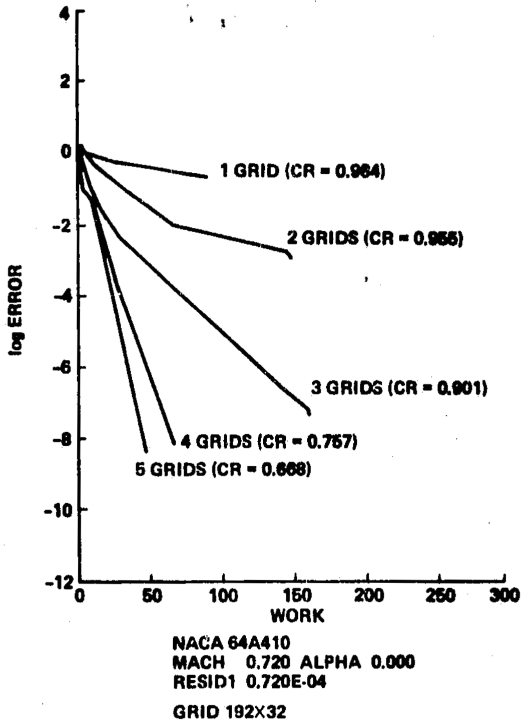
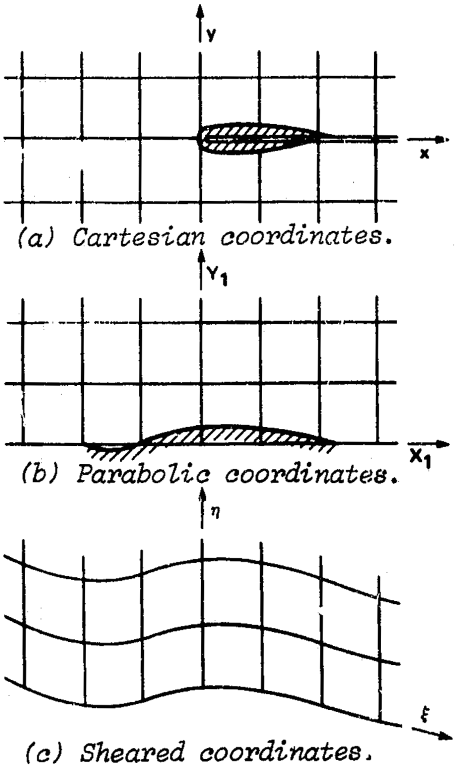


Fig. 35. Multigrid convergence histories, NACA 64A410 airfoil, $M_\infty = 0.72$, $\alpha = 0°$ (ref. 147).



(a) Cartesian coordinates.

(b) Parabolic coordinates.

(c) Sheared coordinates.

Fig. 36. Construction of the sheared parabolic coordinate system used in FL022 (ref. 173).

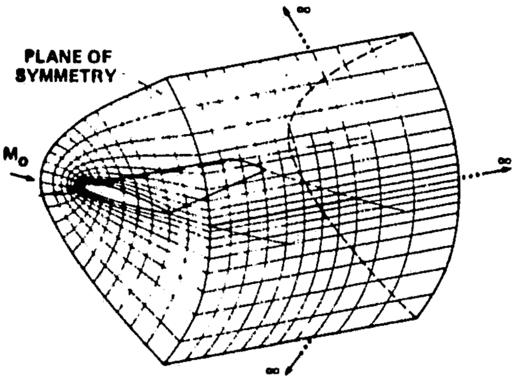


Fig. 37. Sheared parabolic coordinate system used in FL022 (ref. 189).

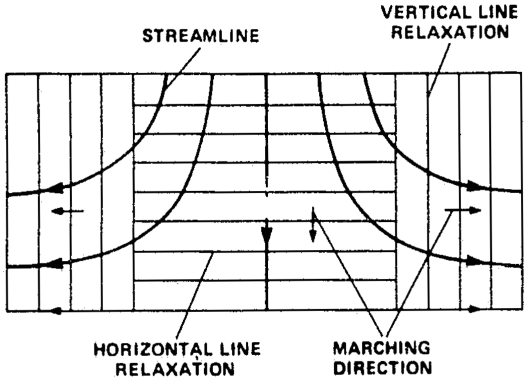


Fig. 38. Marching directions of relaxation scheme used in FL022 (ref. 173).

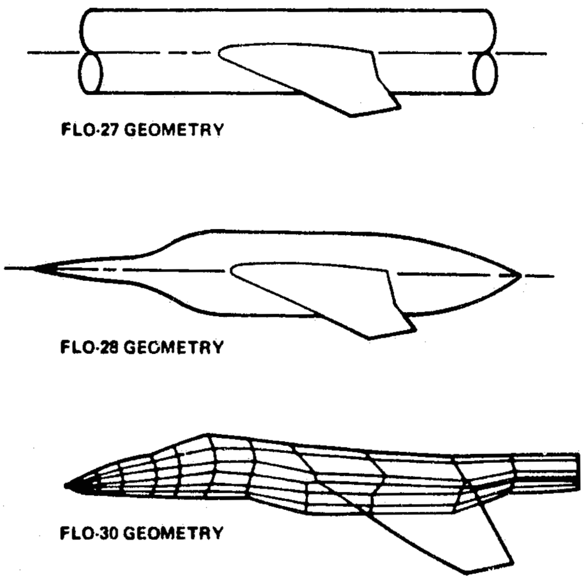


Fig. 39. Comparison of the various FLO code geometry modeling capabilities (ref. 190).

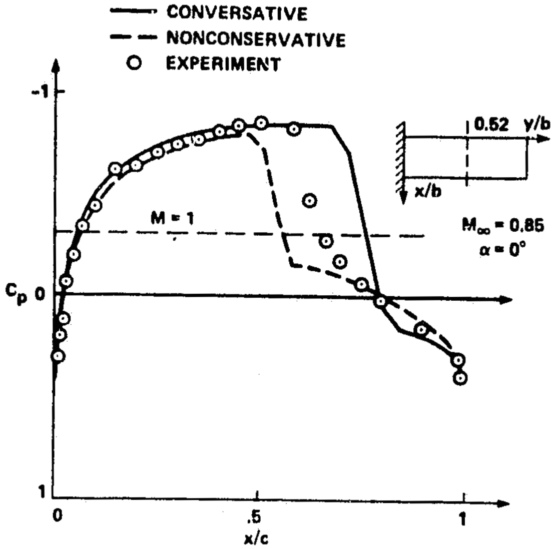
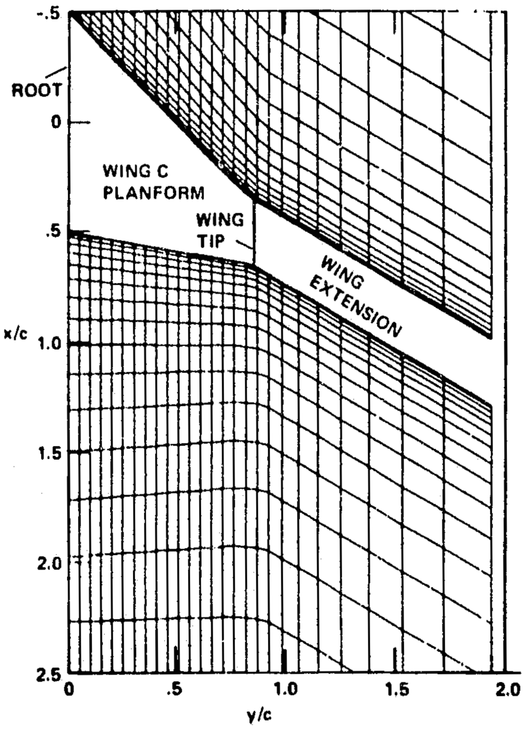
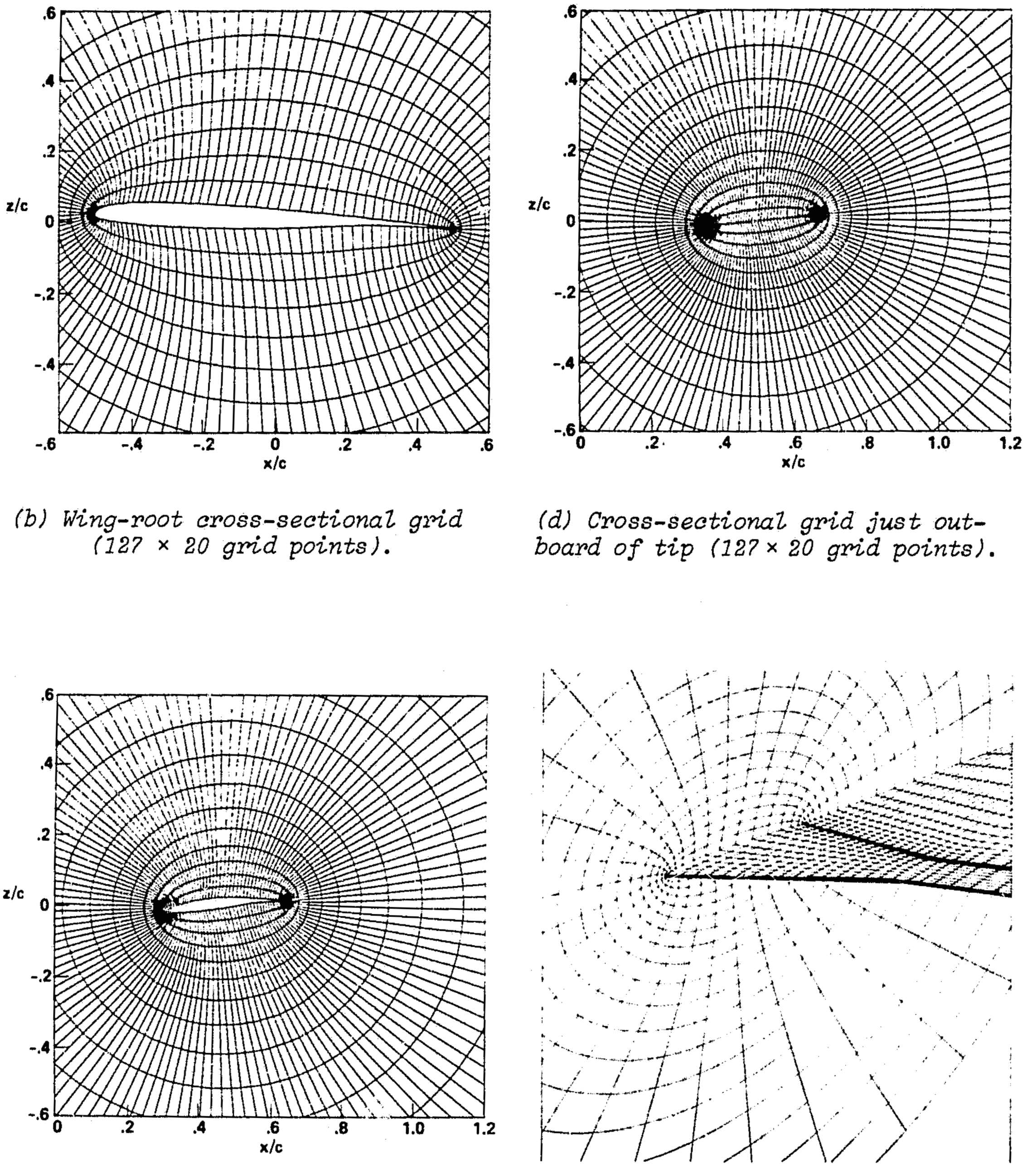


Fig. 40. Pressure coefficient comparisons for a rectangular planform NACA 0012 wing, conservative versus nonconservative differencing (ref. 107).



(a) Wing planform.

Fig. 41. Numerically generated finite-difference mesh about the Hinson-Burdges Wing C configuration (ref. 191).



(b) Wing-root cross-sectional grid (127 × 20 grid points).

(d) Cross-sectional grid just outboard of tip (127 × 20 grid points).

(c) Cross-sectional grid just inboard of the wing tip (127 × 20 grid points).

(e) Perspective view of the grid, 127 × 27 × 20 grid points (only every fourth point plotted in the wraparound direction).

Fig. 41. Concluded.

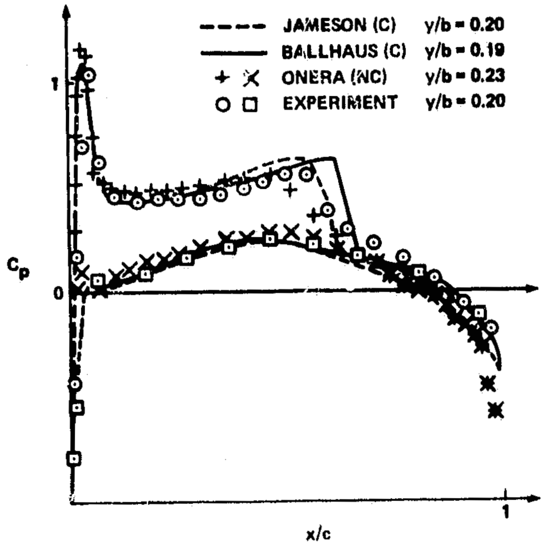


Fig. 42. Inviscid pressure coefficient comparisons for the ONERA M6 wing configuration, $M_\infty = 0.841$, $\alpha = 3^\circ$ (ref. 107).

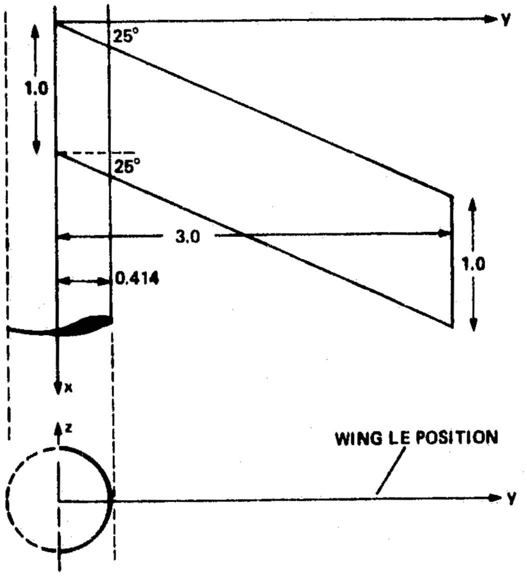
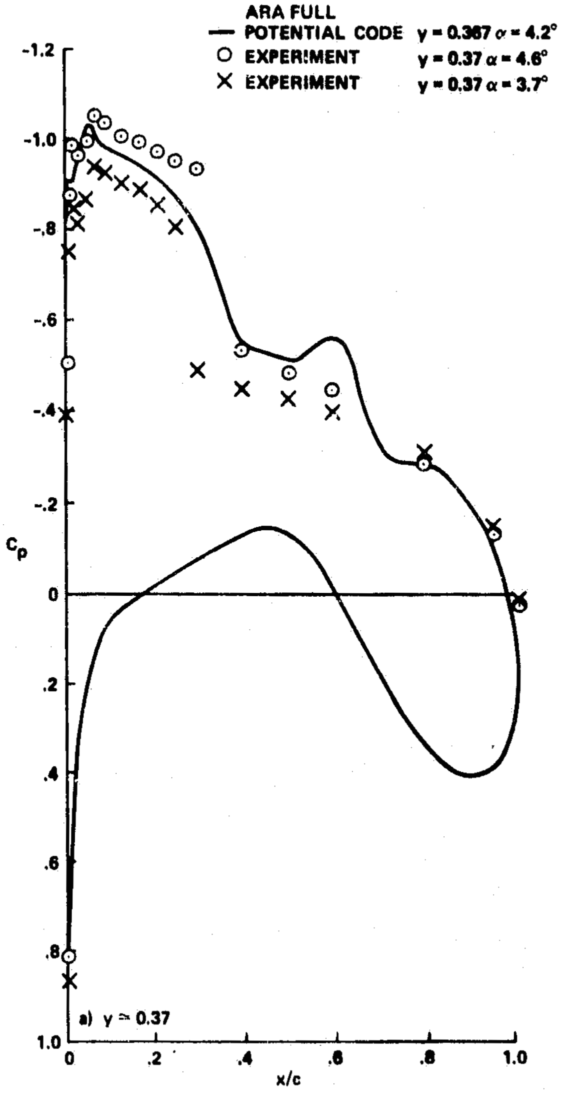
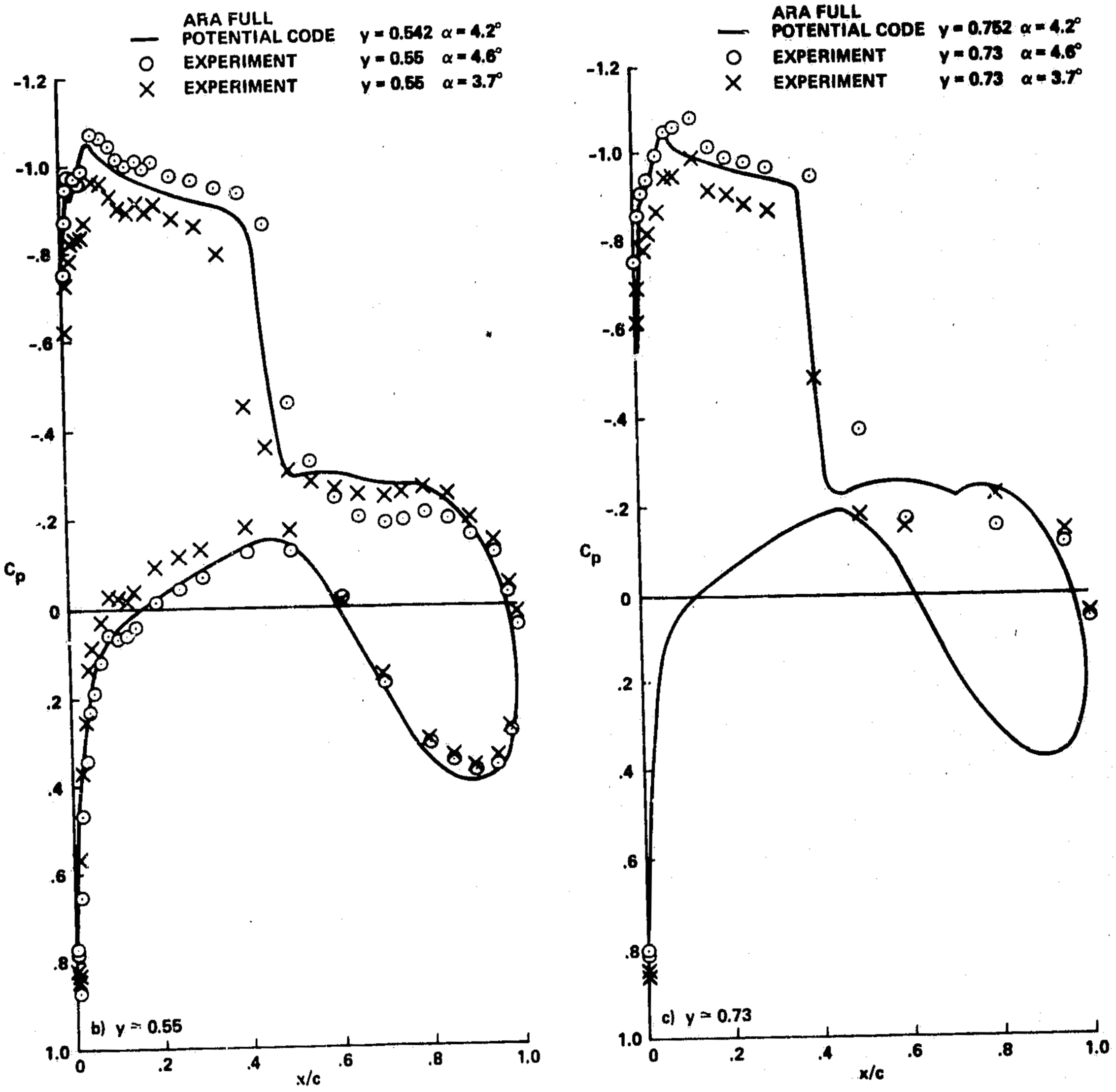


Fig. 43. Wing/body configuration details used for the results presented in Fig. 44 (ref. 2).



(a) y ≅ 0.37.

Fig. 44. Pressure coefficient comparisons for the configuration of figure 43, $M_\infty = 0.86$ (ref. 2).



(b) y ≅ 0.55.

(c) y ≅ 0.73.

Fig. 44. Concluded.

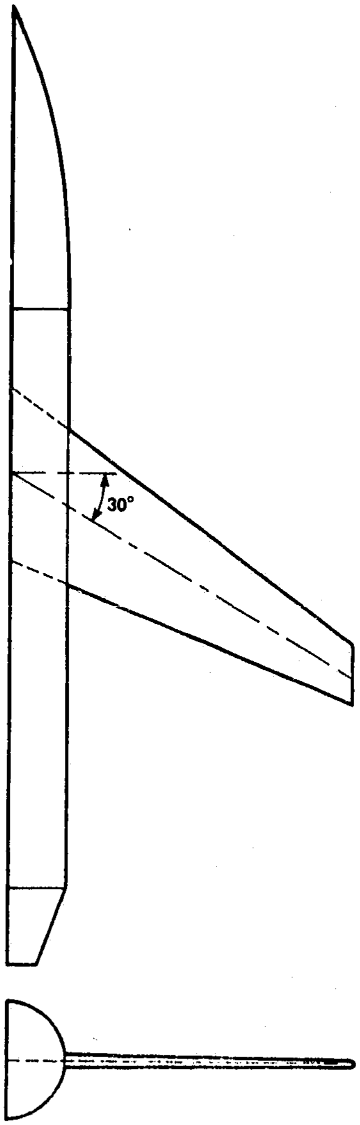


Fig. 45. Wing/fuselage details used for the results presented in figures 46 and 47, RAE wing A + body B2 (ref. 137).

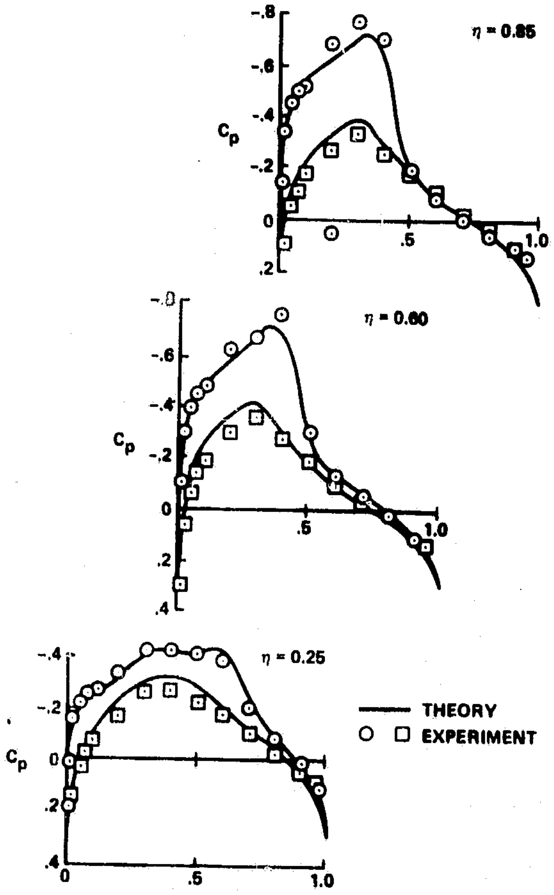


Fig. 46. Pressure coefficient comparison for the configuration of figure 45, $M_\infty = 0.9$, $\alpha = 1°$ (ref. 137).

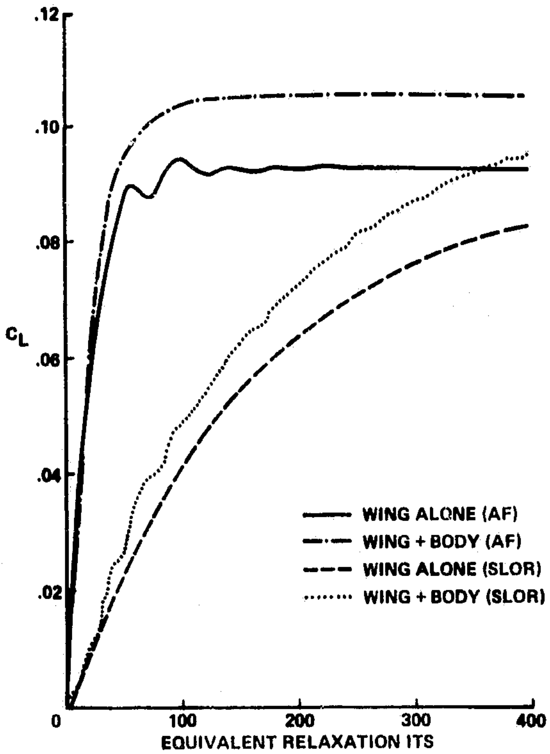
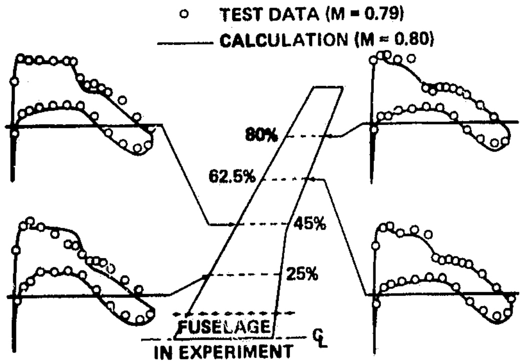


Fig. 47. The development of lift with iteration number for the configuration of figure 45, $M_\infty = 0.9$, $\alpha = 1°$ (ref. 137).



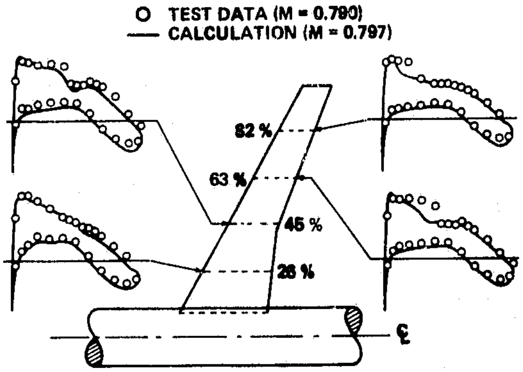


Fig. 48. Comparison of FL022 results with experiment for NASA supercritical wing, Re = 2.4 million (ref. 195).

Fig. 49. Comparison of FL027 results with experiment for NASA supercritical wing, Re = 2.4 million (ref. 195).

ORIGINAL PAGE IS
OF POOR QUALITY

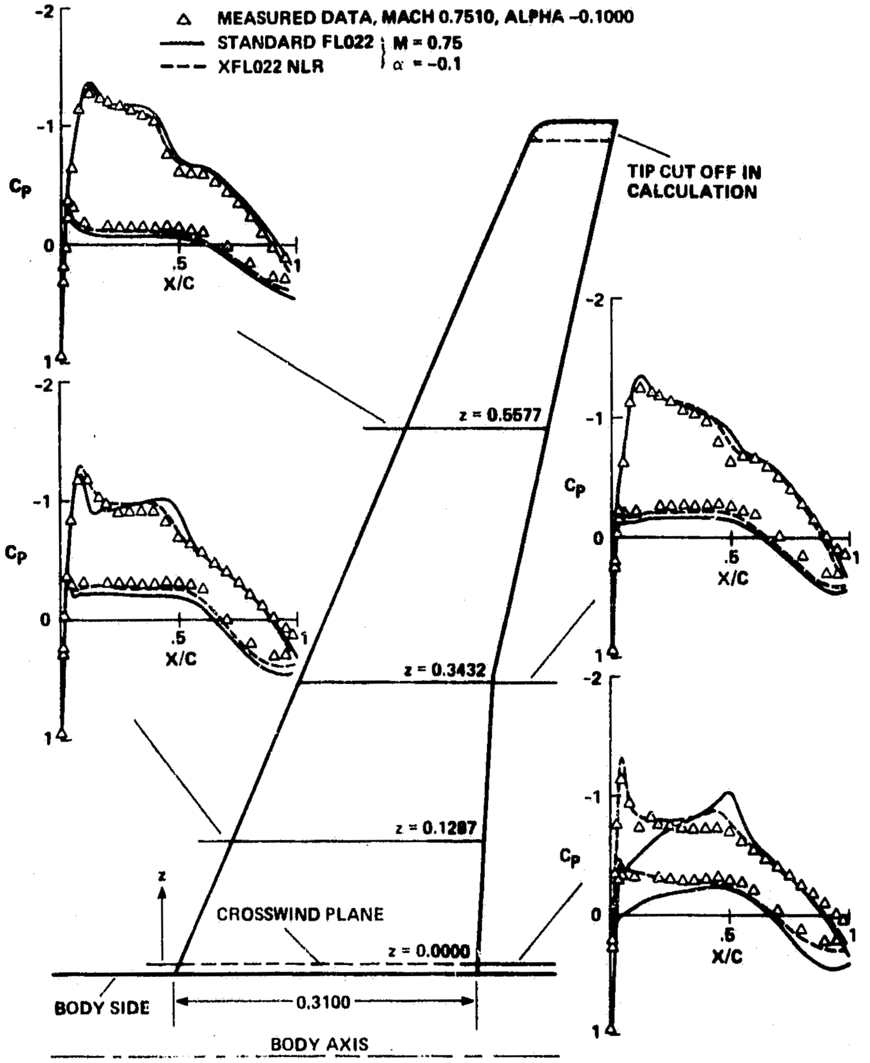


Fig. 50. Body-plus-fillet effect simulation in FL022 by X-wind from panel method (ref. 197).

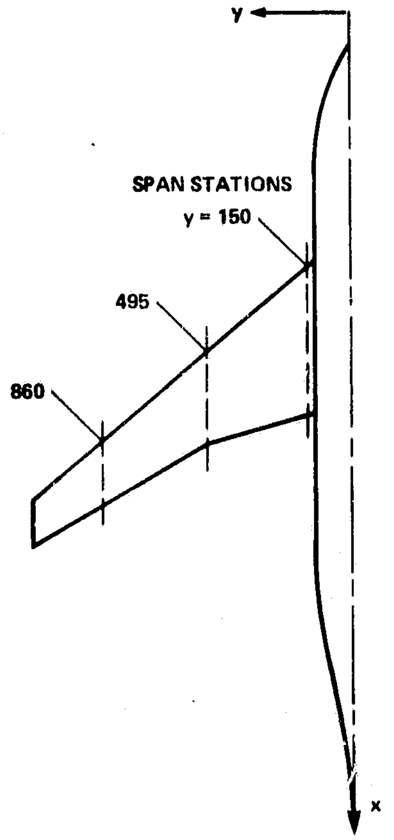
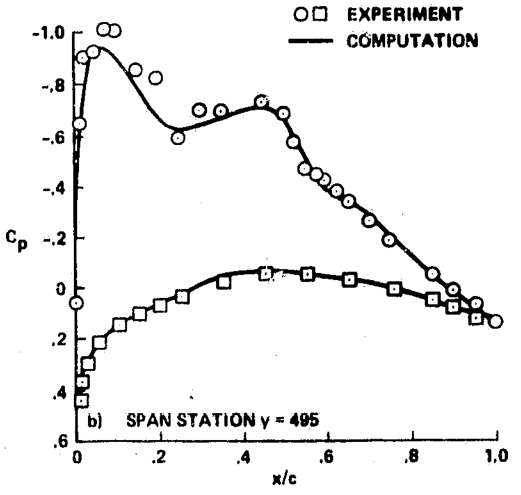
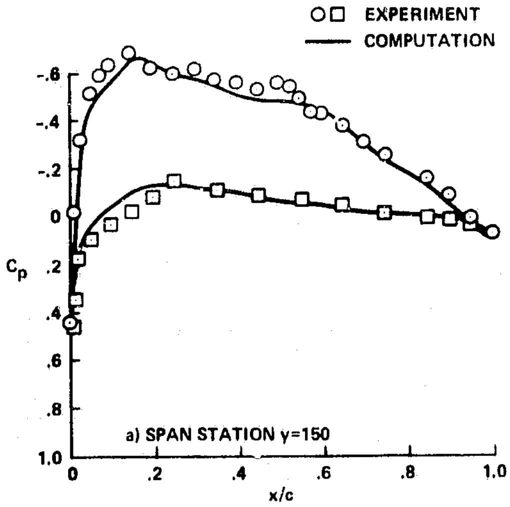


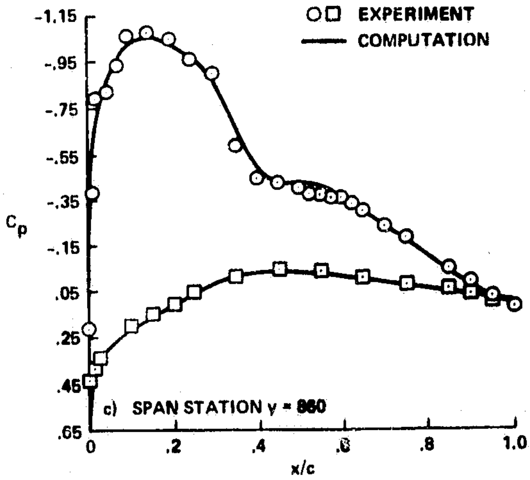
Fig. 51. Boeing 747-200 wing/fuselage configuration used to obtain the results of figure 52 (ref. 2).



(b)



(a)



(c)

Fig. 52. Pressure coefficient comparisons for the configuration of figure 51, M_∞ = 0.84, α = 2.8°, with boundary layer (ref. 2).

ORIGINAL PAGE IS
OF POOR QUALITY

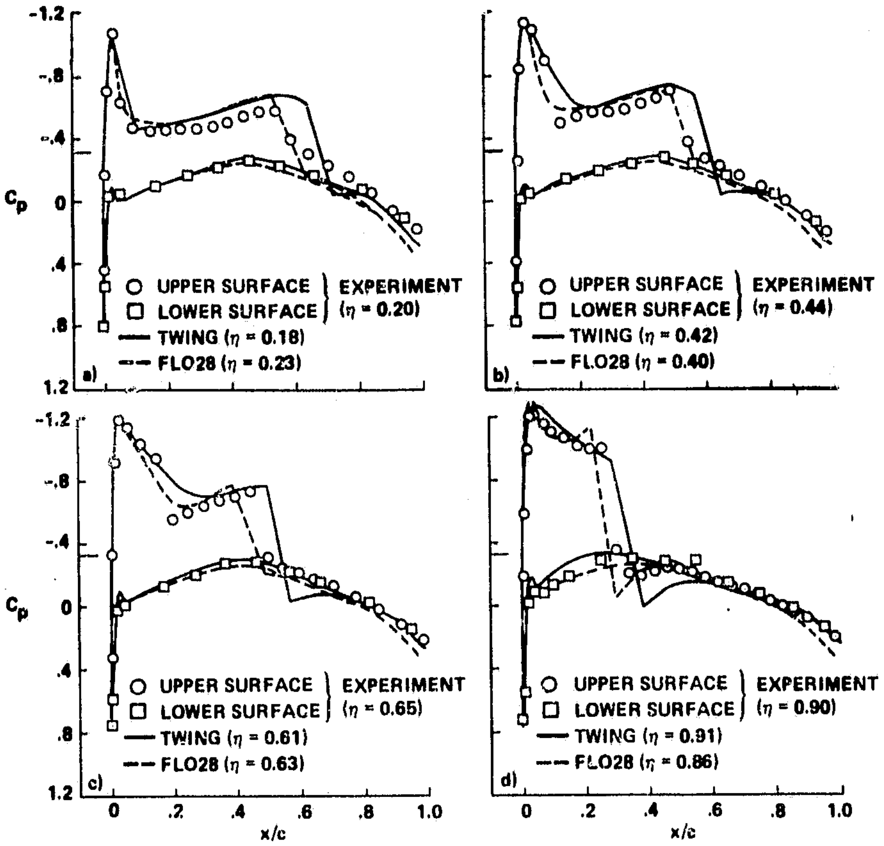
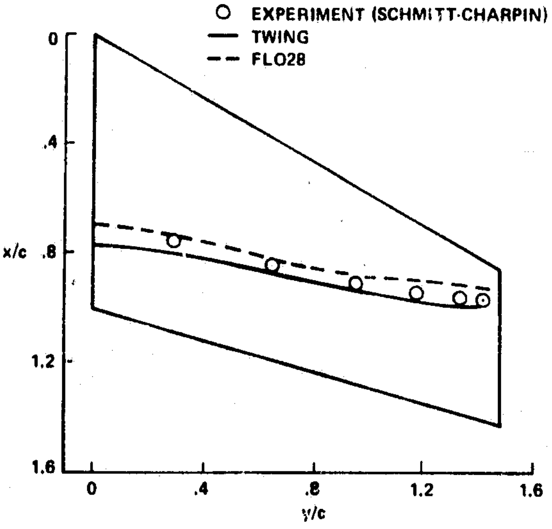


Fig. 53. Pressure coefficient comparisons for the ONERA M6 wing, M_∞ = 0.84, α = 3.06° (ref. 133).



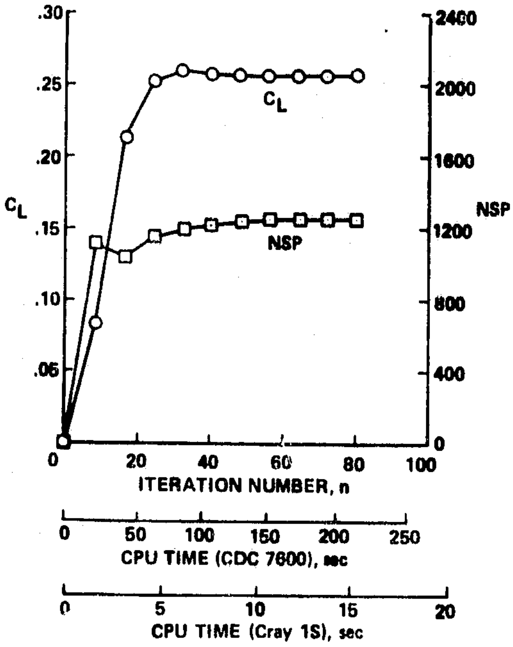


Fig. 54. Comparison of shock sonic line positions for the ONERA M6 wing, M_∞ = 0.84, α = 3.06° (ref. 133).

Fig. 55. Convergence characteristics of TWING for the ONERA M6 wing, M_∞ = 0.84, α = 3.06° (ref. 133).

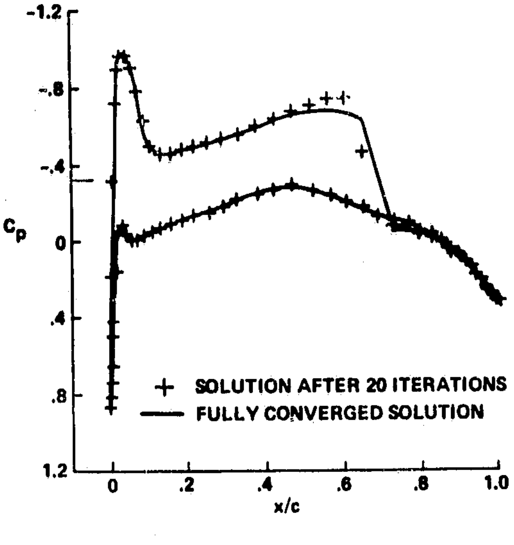


Fig. 56. Comparison of partially and fully converged pressure coefficient distributions at $\eta = 0.20$*, ONERA M6 wing,* $M_\infty = 0.84$*,* $\alpha = 3.06°$ *(ref. 133).*

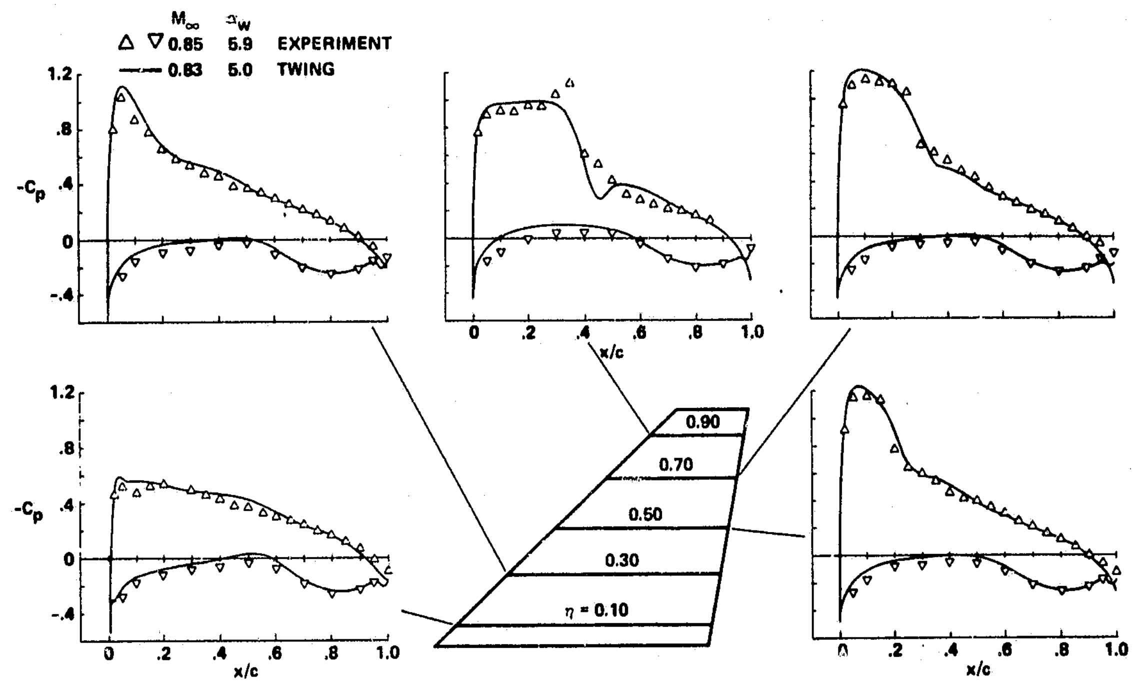
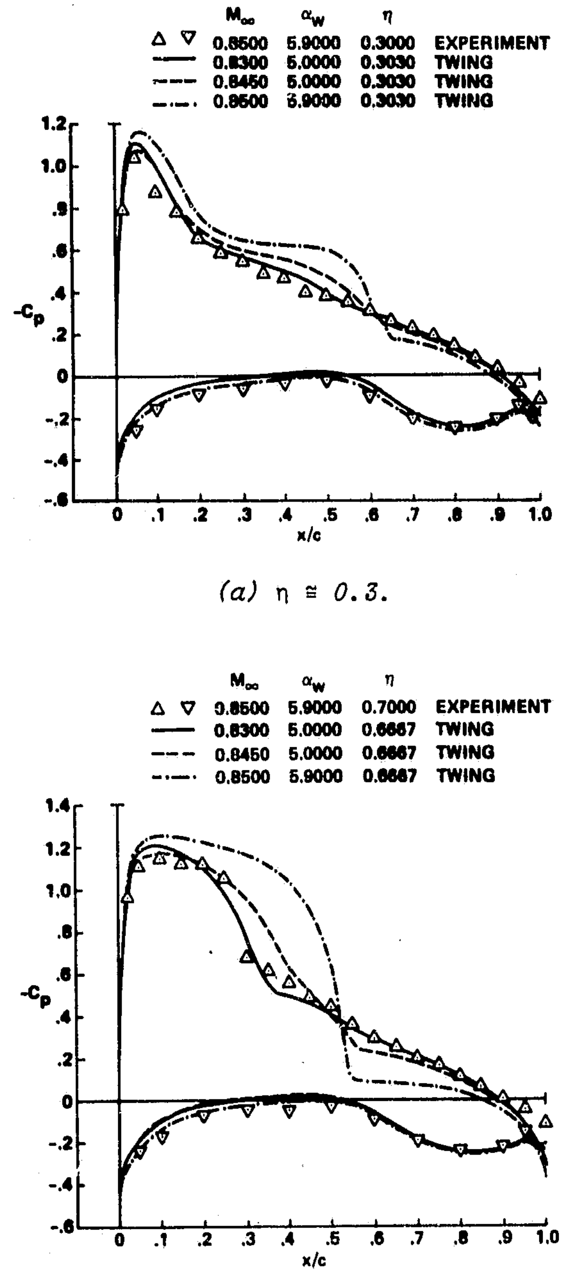


Fig. 57. Comparison of TWING pressure distribution with experiment, Hinson-Burdges Wing C (ref. 191).



(a) η ≅ 0.3.

(b) η ≅ 0.7.

Fig. 58. Effect of Mach number and angle-of-attack corrections on the Hinson-Burdges Wing C pressure coefficient distribution (ref. 191).

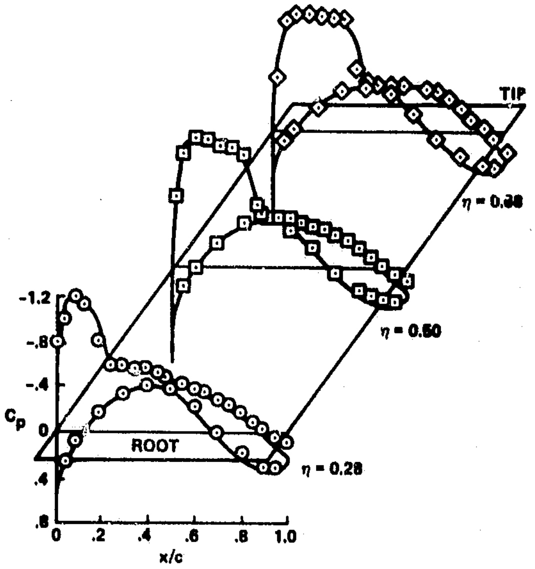


Fig. 59. Comparison of FL030 pressure coefficient distribution with experiment, three-dimensional viscous corrections including wake with thickness and curvature effects modeled, Hinson-Burdges Wing A, M∞ = 0.819, α = 1.96° (ref. 199).

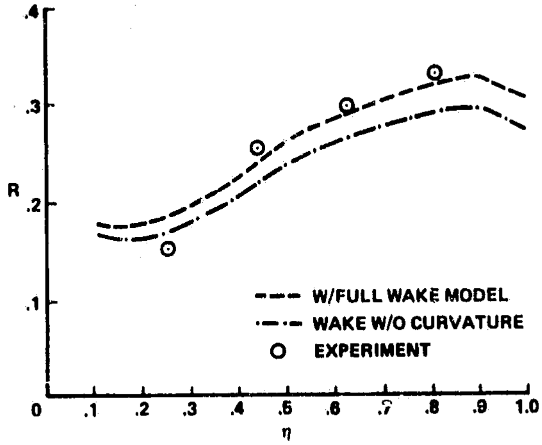


Fig. 60. Lift reduction owing to viscous effects, R, versus semispan station, advanced transport configuration of reference 196 (ref. 199).