



PERGAMON

Progress in Aerospace Sciences 36 (2000) 1-61

PROGRESS IN
AEROSPACE
SCIENCES

www.elsevier.com/locate/paerosci

Transonic flow computations using nonlinear potential methods

Terry L. Holst^{*,1}

NASA Ames Research Center, Moffett Field, Mail Stop T27B-1, CA 94035-1000, USA

Contents

1. Introduction	3
2. Nonlinear potential governing equations	4
2.1. General	4
2.2. Full or exact potential equation	5
2.3. Boundary conditions and circulation	7
2.4. Nonconservative form	8
2.5. Transonic small disturbance (TSD) potential equation	8
2.6. Small-disturbance boundary conditions	9
2.7. Mathematical properties	10
2.8. Transformation techniques	11
2.9. Shock wave capture criteria	13
2.10. Conservative versus nonconservative forms	14
2.11. Full potential equation nonuniqueness	16
3. Algorithms and applications	17
3.1. Early TSD potential equation algorithms and applications	17
3.2. Early full potential equation algorithms and applications	20
3.3. Conservative full potential equation algorithms	21
3.3.1. Finite-volume schemes	21
3.3.2. Artificial density schemes	23
3.3.3. Flux upwind schemes	24
3.3.4. Entropy and vorticity corrections	24
3.3.5. Freestream consistency conditions	25
3.4. Approximate factorization iteration schemes	26
3.5. Convergence characteristics of SLOR, ADI and AF2	28
3.6. Multigrid iteration schemes	30
3.7. Other iteration schemes	32
3.8. Space marching schemes	32
3.9. Time accurate schemes	35
3.10. Design methods	39
3.10.1. Indirect methods	39
3.10.2. Inverse methods	39
3.10.3. Numerical optimization design (gradient methods)	41
3.10.4. Numerical optimization design (genetic algorithms)	43
3.11. Methods developed for complex geometry applications	43
3.11.1. Zonal grid methods	44
3.11.2. Unstructured grid methods	45
3.11.3. Unstructured Cartesian grid methods	48
4. Concluding remarks	49

¹ Research Scientist, Computational Physics and Simulation Branch.

* Tel.: 001-650-604-6032.

E-mail address: tholst@mail.arc.nasa.gov (T.L. Holst).

Nomenclature			
Acronyms			
AF	approximate factorization	x, y, z	Cartesian coordinates (physical domain)
AF2	approximate factorization (scheme 2) introduced by Ballhaus and Steger [1]	x_2, x_η, \dots	physical domain transformation metrics
ADI	alternating direction implicit	α	angle of attack ($^\circ$) or iteration scheme acceleration parameter
CAD	computer-aided design	$\delta_\xi, \delta_\eta, \dots$	central first-difference operators in the ξ, η, \dots directions, respectively
CFD	computational fluid dynamics	$\bar{\delta}_\xi, \bar{\delta}_\eta, \dots$	backward first-difference operators in the ξ, η, \dots directions, respectively
CPU	central processing unit	$\bar{\delta}_\xi, \bar{\delta}_\eta, \dots$	forward first-difference operators in the ξ, η, \dots directions, respectively
FEM	finite element method	Δs	change in entropy across a shock wave
GA	genetic algorithm	ϕ	full or exact velocity potential
GM	gradient method	φ	small-disturbance velocity potential
MFL OPS	million floating-point operations per second	γ	ratio of specific heats = 1.4 (for air)
NSP	number of supersonic points	ρ	fluid density
PDE	partial differential equation	Γ	circulation
SIP	strongly implicit procedure introduced by Stone [2]	Ω	arbitrary closed control volume
SLOR	successive line overrelaxation	θ	angular coordinate in polar coordinate system (rad)
SOR	successive overrelaxation	ξ, η, ζ	spatial coordinates in computational domain
Notation		ξ_x, ξ_η, \dots	computational domain transformation metrics
TSD	transonic small disturbance	τ	time in computational domain
a	speed of sound	∇^2	Laplacian operator
$A_1 - A_6$	transformation metrics	∇	gradient operator
c	wing chord	∂	partial derivative operator
C_L	lift coefficient		
C_p	pressure coefficient		
D/Dt	material or substantial derivative		
i, j, k	unit vectors along x, y, z	<i>Subscripts</i>	
J	determinant of independent variable transformation Jacobian	i, j, k	grid indices corresponding to ξ, η, ζ computational coordinates
L	lift	x, y, z	indicates differentiation with respect to x, y, z
M_∞	freestream Mach number	ξ, η, ζ	indicates differentiation with respect to ξ, η, ζ
\mathbf{n}	unit outward normal vector	∞	freestream condition
p	fluid pressure		
s, n	stream and stream-normal coordinates		
S	boundary surface surrounding Ω		
t	physical time coordinate	<i>Superscripts</i>	
u, v, w	velocity components along x, y, z	n	iteration index (n th iteration)
q	velocity vector magnitude	*	critical condition (condition at $M = 1$)
\mathbf{q}	velocity vector		
t	physical time or time-like coordinate		

5. Recommendations for future work	51
Acknowledgements	52
References	52

Abstract

This presentation describes the state of transonic flow simulation using nonlinear potential methods for external-aerodynamic applications. The presentation begins with a review of the various potential equation forms (with emphasis on the full potential equation) and includes a discussion of pertinent mathematical characteristics and all derivation assumptions. Impact of the derivation assumptions on simulation accuracy, especially with respect to shock wave capture, is discussed. Key characteristics of all numerical algorithm types used for solving nonlinear potential equations, including steady, unsteady, space marching, and design methods, are described. Both spatial discretization and iteration scheme characteristics are examined. Numerical results for various aerodynamic applications are included throughout the presentation to highlight key discussion points. The presentation ends with concluding remarks and recommendations for future work. Overall, nonlinear potential solvers are efficient, highly developed and routinely used in the aerodynamic design environment for cruise conditions. Published by Elsevier Science Ltd. All rights reserved.

1. Introduction

In the past three decades the field of computational aerodynamics has evolved from a curious art barely capable of predicting the inviscid flow over simple two-dimensional shapes, such as airfoils, to the relatively mature current capability, which is capable of predicting high-Reynolds-number viscous flows about complex three-dimensional shapes, including nearly complete aircraft. The development of this scientific/engineering field has been paced by many milestones in a variety of areas, including computational hardware, system software, computing environments, numerical algorithms, computer graphics, geometric modeling, flow solver algorithms, etc. Indeed, many volumes could be written in describing the development of computational aerodynamics. The purpose of this review is to describe a small, and yet very important aspect of computational aerodynamics, that portion associated with nonlinear potential formulations. This area is important because numerical simulations based on nonlinear potential equations provide quantitative answers to aerodynamic questions in a small amount of wall clock time. For aircraft design, reducing wall clock time is vitally important because it means lower development costs and rapid product availability, which both contribute to larger market share.

Potential equation numerical simulations are computationally efficient because they involve the solution of a simple scalar equation. The more complete formulations of computational fluid dynamics (CFD), the Navier–Stokes and Euler equations, consist of five coupled equations. In addition, numerical iteration schemes for solving the potential equations typically converge in fewer iterations than iteration schemes for the Euler or Navier–Stokes equations. Thus, potential solvers are typically an order of magnitude (or more) faster than Euler equation solvers on comparable grids [3]. The price for this extraordinary speed is limitation of application. All potential formulations are inherently irrotational and isentropic. These assumptions are generally consistent with subsonic, transonic and supersonic flows at or near cruise conditions providing all shock waves are weak. If strong shock waves exist in the flow field, i.e., shocks with an upstream, normal-shock Mach number component at or above about 1.3, then the full potential solution will be in error; the stronger the shock wave, the larger the error. A major ameliorating characteristic of this situation is that for cruise conditions (associated with the transonic flow regime), the existence of strong shock waves is a very undesirable characteristic. If a candidate configuration has a strong shock wave, a numerical result does not have to be very accurate to eliminate it from further consideration in the design process. Ideally, as the configuration is refined, the shock strength is reduced and the full potential equation accuracy is improved. This is why the full potential formula-

tion is used so much in aerodynamic shape design for transonic cruise conditions.

The main emphasis of this review is to describe numerical solution techniques for solving transonic flow problems governed by the full potential equation. Because algorithms for solving the transonic small disturbance (TSD) potential equation are very similar in nature, this topic is covered as well, but in less detail. 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.

The transonic flow regime provides the most efficient aircraft cruise performance; hence, most large commercial aircraft cruise in this 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. For more information on how CFD methods, in general, are being used in (and have benefited) the aircraft design environment, the interested reader is referred to Rubbert [4].

Transonic flow fields contain a variety of interesting and unique characteristics. Typical airfoil and swept-wing flow fields are shown in Figs. 1 and 2. The outer freestream flow is typically subsonic and elliptic in nature. Regions of supersonic flow 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 Fig. 2. The first shock is swept and therefore has a supersonic downstream Mach number. The aft 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 speed is $u + a$ (local flow speed plus speed of sound) and very slowly upstream where the propagation speed 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

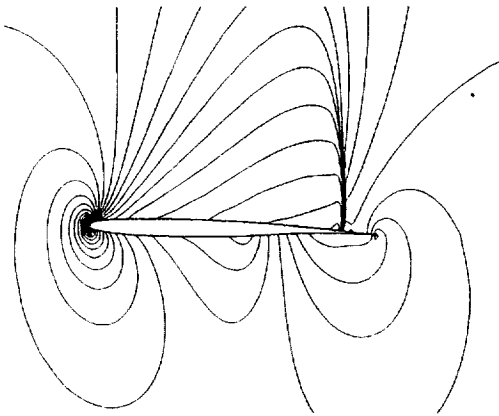


Fig. 1. Mach number contours about an airfoil showing a typical two-dimensional transonic, inviscid flow field computed using a full potential algorithm.

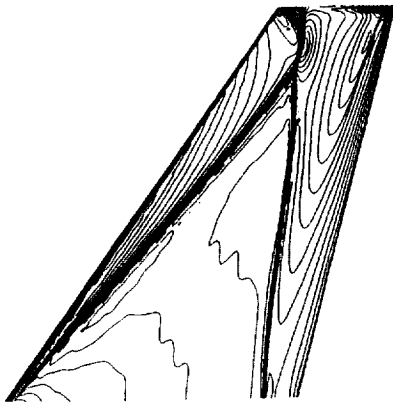


Fig. 2. Mach number contours on the upper surface of a swept wing showing a typical three-dimensional transonic, inviscid flow field computed using a full potential algorithm.

physical-time-dependent algorithms, very slow (relative to similar algorithms for subsonic or supersonic flow problems). Such problems are said to be “stiff,” and require larger amounts of computer time.

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 is responsible for the prediction of shock waves. In contrast, inviscid subsonic flow can be linearized with good accuracy. The result is Laplace’s equation (or a relative thereof), which can be solved using a direct method, i.e., a method without iteration. The inherent nonlinear behavior of transonic flow problems means that a direct solution is impossible. Thus, one basic feature associated with all transonic-flow numerical schemes is that they must be iterative.

Viscous effects are also extremely important in transonic flows. 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. Although a discussion of viscous correction procedures is not within the scope of this review, an ample number of references are presented for those potential solvers that have viscous correction procedures included.

This review begins with a discussion of the various nonlinear potential formulations that have been utilized in the field of computational transonic aerodynamics over the past two or three decades (chapter 2). Formulation assumptions and limitations, nonconservative versus conservative forms, shock capturing capabilities and nonuniqueness issues are discussed in detail in this chapter. Next, in chapter 3, the presentation continues with a review of past and present research activities involving algorithm development and aerodynamic applications with primary emphasis on the full potential formulation. This chapter includes the milestone achievements that have shaped the current state of the art in transonic potential methods. Solution methods reviewed include classical relaxation algorithms, time-accurate schemes, supersonic space marching schemes and design methods. Numerical result examples are included throughout chapter 3 to highlight important discussion points. The presentation ends with concluding remarks (chapter 4) and recommendations for future work (chapter 5).

Many additional review papers on this and other related topics are available. A few of these include Hall [5] and South [6] where a historical development of the potential formulation in computational aerodynamics is presented; Holst et al. [7] Kordulla [8] and Nixon and Kerlick [9] where a variety of transonic potential flow simulation surveys are presented; and the collected papers in Nixon [10] Caughey and Hafez [11] Zierp and Oertel [12] Habashi [13] and Henne [14] where a wealth of information about the more general topics of computational aerodynamics and transonic aerodynamics are presented. Finally, additional basic information about numerical solution algorithms for nonlinear potential formulations is available in Hirsch [15] Anderson et al. [16] and Pai and Luo [17].

2. Nonlinear potential governing equations

2.1. General

There are several different potential equation formulations used in aerodynamic simulations. Although this presentation deals primarily with the full or exact velocity potential formulation, it is of interest to review all potential formulations to establish differences and

similarities. All potential flow formulations are based on the ability to define a velocity potential, which requires an irrotational flow assumption. Thus, for a velocity vector field defined by \mathbf{q} , the requirement for a velocity potential to exist is

$$\nabla \times \mathbf{q} = 0. \quad (1)$$

If this condition holds in all locations of the flow field of interest, then a full or exact velocity potential function ϕ exists and is defined by

$$\nabla \phi = \mathbf{q}. \quad (2)$$

Velocity components can be expressed in terms of partial derivatives of the velocity potential function. For Cartesian coordinates this is given as

$$\nabla \phi = \mathbf{q} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} = \phi_x\mathbf{i} + \phi_y\mathbf{j} + \phi_z\mathbf{k}$$

or component by component as

$$\phi_x = u, \quad \phi_y = v, \quad \phi_z = w,$$

where \mathbf{i} , \mathbf{j} and \mathbf{k} are the standard unit vectors in the x , y and z directions, respectively; and u , v and w are the Cartesian velocity vector components, also along the x , y and z directions, respectively. In the above expressions, the quantity ϕ_x (for example) is used to indicate a partial derivative of ϕ with respect to the spatial coordinate x .

The velocity potential function has a spatial variation which is independent of path. It can be defined for incompressible or compressible flows that are either steady or unsteady. It is restricted, however, because of the irrotational-flow assumption, to flows without viscous effects, i.e., potential flows are inherently inviscid in nature. Of course, viscous boundary layer corrections can be included quite easily (at least for attached flows) by solving a potential formulation in conjunction with the boundary layer equations, but this is beyond the scope of this presentation and will not be discussed further. Other types of rotational flow corrections can be included, e.g., flows with circulation and/or vortices, but this requires additional (often empirical) modeling. More on circulation modeling will be presented in the section on boundary conditions. Next, the discussion turns to the various governing equation forms that utilize a velocity potential function.

2.2. Full or exact potential equation

The most general form of the full potential equation is derived from the mass continuity equation using the definition of the velocity potential given above [Eq. (2)]. This equation, written in integral form, is given by

$$\frac{\partial}{\partial t} \iiint_{\Omega} \rho \, d\Omega + \iint_S \mathbf{n} \cdot \rho \nabla \phi \, dS = 0, \quad (3)$$

where t is the physical time coordinate, ρ is the fluid density, Ω is an arbitrary closed control volume, S is the boundary surface surrounding Ω , and \mathbf{n} is the unit outward normal vector to the surface S . Eq. (3) states that the time rate of change of the mass in an arbitrary fixed volume (first term) is balanced by the net outflow of mass leaving the same volume (second term). In order for Eq. (3) to represent a closed-form description of a flow field, an algebraic expression for the density in terms of the velocity potential must be utilized. Several forms of this expression will be discussed shortly.

Eq. (3) can be expressed in differential form by transforming the surface integral term into a volume integral using Gauss' Divergence Theorem:

$$\iint_S \mathbf{n} \cdot \rho \nabla \phi \, dS = \iiint_{\Omega} \nabla \cdot \rho \nabla \phi \, d\Omega.$$

Using the fact that the control volume Ω is fixed with respect to time, the differentiation and integration associated with the first term of Eq. (3) can be interchanged. Combining the two volume integrals into the same term, Eq. (3) becomes

$$\iiint_{\Omega} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \nabla \phi \right) d\Omega = 0.$$

Since the control volume is arbitrary, the integrand in the above equation vanishes everywhere, which results in the desired differential form of the unsteady full potential equation.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \nabla \phi = 0. \quad (4)$$

The above integral and differential forms of the full potential equation still need an additional relation to complete the formulation. In particular, a relation that expresses the fluid density ρ as a function of the velocity components ϕ_x , ϕ_y and ϕ_z , is required. A suitable approach for this derivation starts with the inviscid momentum equation given by (this is actually a form of the Euler momentum equation)

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbf{q} \cdot \nabla \mathbf{q} + \frac{\nabla p}{\rho} = 0,$$

where p is the fluid pressure. The second term in the above equation can be reduced to a convenient form using the Lagrange acceleration formula [18] given by

$$\frac{D\mathbf{q}}{Dt} = \frac{\partial \mathbf{q}}{\partial t} + \nabla \left(\frac{q^2}{2} \right) - \mathbf{q} \times (\nabla \times \mathbf{q}),$$

where q in the second term on the right-hand side represents the magnitude of the velocity vector. The D/Dt notation used on the left-hand side stands for the

material or substantial derivative defined by

$$\frac{D\mathbf{q}}{Dt} = \frac{\partial\mathbf{q}}{\partial t} + \mathbf{q} \cdot \nabla\mathbf{q}.$$

For irrotational fluids the following equation is easily obtained

$$\mathbf{q} \cdot \nabla\mathbf{q} = \nabla\left(\frac{q^2}{2}\right).$$

Using this relation the momentum equation becomes

$$\frac{\partial\mathbf{q}}{\partial t} + \nabla\left(\frac{q^2}{2}\right) + \frac{\nabla p}{\rho} = 0.$$

Substituting $\nabla\phi$ for \mathbf{q} and using the fact that

$$\nabla\int\frac{dp}{\rho} = \frac{\nabla p}{\rho}$$

for a barotropic fluid² yields

$$\nabla\left(\frac{\partial\phi}{\partial t} + \int\frac{dp}{\rho} + \frac{q^2}{2}\right) = 0.$$

The integration of this expression along an arbitrary line in the flow domain yields

$$\frac{\partial\phi}{\partial t} + \int\frac{dp}{\rho} + \frac{q^2}{2} = C(t), \quad (5)$$

where $C(t)$ is a constant of integration that in general is a function of time but not space. Eq. (5) is the unsteady Bernoulli equation. It and various related forms of the Bernoulli equation are used in many areas of fluid dynamics.

The integral in the unsteady Bernoulli equation must be evaluated before this equation can be used further. This is accomplished using

$$\frac{p}{\rho^\gamma} = \text{const}, \quad (6)$$

where γ is the ratio of specific heats (equal to 1.40 for air) and “const” is a constant that can be evaluated when a nondimensionalization is chosen. Eq. (6) is the standard density–pressure relationship for an isentropic flow. With this relation the integral in Eq. (5) can easily be evaluated yielding

$$\int\frac{dp}{\rho} = \frac{\gamma}{\gamma-1}\frac{p}{\rho} = \frac{a^2}{\gamma-1},$$

where a is the fluid speed of sound. The last equality in the above equation is obtained using the perfect gas

speed of sound definition. Thus, the final form of the integrated unsteady Bernoulli equation is given by

$$\frac{\partial\phi}{\partial t} + \frac{a^2}{\gamma-1} + \frac{q^2}{2} = C(t). \quad (7)$$

The ultimate goal of this derivation is obtaining a relationship between the fluid density and derivatives of the velocity potential, i.e., the fluid’s velocity components, thus allowing closure of the full potential equations given above by Eqs. (3) or (4). This is accomplished using Eq. (7), the speed of sound definition, and the isentropic density–pressure relation [Eq. (6)], yielding the final desired relation

$$\rho = \left[1 + \frac{\gamma-1}{2}(M_x^2 - 2\phi_t - \phi_x^2 - \phi_y^2 - \phi_z^2)\right]^{1/(\gamma-1)}. \quad (8)$$

In this equation the density ρ and the velocity components ϕ_x , ϕ_y and ϕ_z are nondimensionalized by the freestream values of the density ρ_∞ and the freestream value of the speed of sound a_∞ , respectively. The Cartesian coordinates x , y and z and the time t are nondimensionalized by a characteristic length, usually the airfoil or wing chord c and the quantity a_∞/c , respectively.

Several other forms of the full potential equation with different types of nondimensionalization have been used for numerical computations. For example, the steady flow version of the differential form of the full potential equation in which the density is nondimensionalized by the stagnation density ρ_{stag} and all velocity components are nondimensionalized by the critical speed of sound a^* is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0, \quad (9)$$

$$\rho = \left[1 - \frac{\gamma-1}{\gamma+1}(\phi_x^2 + \phi_y^2 + \phi_z^2)\right]^{1/(\gamma-1)}. \quad (10)$$

All length scales are still nondimensionalized by the same characteristic length. Use of the above nondimensionalization creates the following useful conditions. At stagnation points

$$\rho = 1, \quad \phi_x = \phi_y = \phi_z = 0,$$

and at sonic lines

$$\begin{aligned} \phi_x^2 + \phi_y^2 + \phi_z^2 = 1, \quad \rho &= \left[1 - \frac{\gamma-1}{\gamma+1}\right]^{1/(\gamma-1)} \\ &= 0.633938145 \dots \quad (\gamma = 1.40). \end{aligned}$$

In addition, either of the two nondimensionalizations given above can be used to evaluate the constant in the isentropic density–pressure relation [Eq. (6)] or the constant in the steady Bernoulli equation, which is Eq. (7)

²A barotropic fluid is one in which the density can be expressed solely as a function of pressure, i.e., $\rho = \rho(p)$. For example, a fluid undergoing an isentropic process is a barotropic fluid.

with the time terms removed. For example, using the second nondimensionalization from above (ρ_{stag} and a^*), the following results are obtained:

Isentropic density–pressure relation

$$\frac{p}{\rho^\gamma} = \frac{\gamma + 1}{2\gamma}. \quad (11)$$

Steady Bernoulli's equation:

$$\frac{q^2}{2} + \frac{a^2}{\gamma - 1} = \frac{1}{2} \frac{\gamma + 1}{\gamma - 1}. \quad (12)$$

Another widely used potential equation form is obtained from Eq. (9) by making the additional assumption of incompressible flow. This yields the familiar Laplace's equation

$$\nabla^2 \phi = \phi_{xx} + \phi_{yy} + \phi_{zz} = 0. \quad (13)$$

Unlike the nonlinear full potential equation given by Eqs. (3), (4) or (9), Laplace's equation is linear. Although this further limits the flow field physics that can be simulated, e.g., shocks cannot be captured with Laplace's equation, the linear nature lends itself to the powerful method of superposition. A numerical approach that utilizes superposition to solve Laplace's equation is typically called a panel method. A key panel method characteristic is that only the geometric surface of interest need be discretized. In contrast, all numerical methods used to solve nonlinear governing equations require a field or volume discretization, a feature that greatly increases the numerical algorithm complication. In a panel method, flow tangency along the aerodynamic surface is obtained by solving for source, vortex or doublet distribution strengths for the surface's discretized elements. This operation requires and is computationally paced by the inversion of a large matrix whose rank is equal to the number of boundary elements. Thus, a key aspect of any panel method implementation is the judicious selection (both in number and placement) of an appropriate surface element discretization. Panel method details are beyond the scope of this presentation and will not be discussed further. The interested reader is referred to Anderson et al. [16] for basic information on panel methods; Smith [19] for a historical presentation of panel method development; and Hess [20] Høijmakers [21] and Roggero and Larguier [22] for information on current panel method applications.

2.3. Boundary conditions and circulation

To complete the full potential governing equation specification, boundary conditions are required along all boundaries. Specifically, these boundaries fall into three categories: freestream, symmetry planes, and geometric surfaces. The freestream boundary condition, simply

stated, is given by

$$x^2 + y^2 + z^2 \rightarrow \infty, \quad \phi \rightarrow \phi_\infty,$$

where ϕ_∞ is the freestream distribution of the velocity potential, usually uniform flow. The latter two boundary conditions, symmetry planes and geometric surfaces, are both treated in the same manner, i.e., with a flow tangency assumption given by

$$\mathbf{q} \cdot \mathbf{n} = 0,$$

where \mathbf{n} is a unit vector normal to the geometry of interest. More on flow tangency boundary conditions is presented in Section 2.8 where transformation techniques for the full potential equation are discussed.

For aerodynamic applications to be useful the numerical formulation must be able to predict aerodynamic loads, e.g., lift. The Kutta–Joukowski theorem says

$$L = \rho_\infty q_\infty \Gamma,$$

where L is the lift, ρ_∞ is the freestream fluid density, q_∞ is the freestream fluid velocity magnitude, and Γ is the circulation. The circulation around (for example) an airfoil is mathematically defined as

$$\Gamma = \oint_l \mathbf{q} \cdot d\mathbf{l},$$

where l is any closed path surrounding the airfoil for which the velocity vector field is defined. Using Stokes' Theorem it can be seen that circulation is inherently tied to vorticity, that is

$$\Gamma = \oint_l \mathbf{q} \cdot d\mathbf{l} = \iint_S \nabla \times \mathbf{q} \cdot \mathbf{n} \, dS,$$

where S is the surface constructed such that its boundary is l and \mathbf{n} is the unit outward normal vector to S . Thus, it can be seen from the above equation that an irrotational velocity vector field, such as that predicted by the full potential equation, is not capable of supporting circulation, i.e., there is no lift.

This situation can be corrected by adding a linear potential vortex solution to the nonlinear potential that surrounds the airfoil. This is accomplished by modifying the freestream boundary condition as follows

$$\phi_{\text{ob}} = \phi_\infty + \phi_v,$$

where ϕ_{ob} is the new outer boundary condition, ϕ_∞ is the usual uniform-flow velocity potential solution, and ϕ_v is the newly added potential vortex solution given by

$$\phi_v = \frac{\Gamma}{2\pi} \theta.$$

In the above equation, $\Gamma/2\pi$ is a constant representing the vortex strength and θ is the usual angular coordinate associated with a traditional polar coordinate system centered inside the airfoil. The vorticity associated with the resulting potential solution is zero everywhere except at the center of the vortex where it is infinite. The circulation is a constant (equal to Γ) for all integration curves that include the airfoil and zero for all integration curves that do not include the airfoil. Because of the periodic nature of θ , the ϕ_v function is double-valued at $\theta = 0$, taking on values of 0 and 2π . In other words, along some “cut” in the velocity potential solution, usually emanating from the airfoil trailing edge to downstream infinity along the airfoil wake, the velocity potential “jumps” from its $\theta = 0$ value to its $\theta = 2\pi$ value. The magnitude of this velocity potential jump (easily derived by looking at the definition of ϕ_v) is equal to Γ . In effect, forcing the ϕ_v vortex strength to be equal to the airfoil trailing edge velocity potential jump, is like a Kutta condition that forces the airfoil-surface upper and lower pressures to match.

2.4. Nonconservative form

Eqs. (4) and (9) are two forms of the full potential equation which are commonly used in numerical applications, especially when shock waves are expected to be captured in the solution. These versions of the full potential equation are written in the so-called conservative form, which is characterized by having all variables inside the outer-most differentiation. The steady full potential equation has also been solved in nonconservative form for subsonic and transonic applications. For subsonic applications the nonconservative and conservative solutions, assuming equivalently small levels of numerical error, are virtually identical. For transonic applications involving captured shock waves the nonconservative and conservative solutions are different. At shock waves the nonconservative approach produces an error in the form of a mass source that causes an error in shock position and strength. The consequence of this error is discussed in detail in Section 2.10. The nonconservative form of the full potential equation is presented here because of its historical importance and because it provides a useful framework for analyzing the conservative full potential equation.

The nonconservative form of the full potential equation is derived from the conservative form by using the chain rule to expand derivatives. Expressions for the density derivatives are obtained from the density expression [Eq. (10)], the speed of sound definition, and the isentropic density–pressure relation [Eq. (11)]. Substitution of these derivatives into the expanded full potential equation yields

$$(a^2 - u^2)\phi_{xx} + (a^2 - v^2)\phi_{yy} + (a^2 - w^2)\phi_{zz} - 2uv\phi_{xy} - 2uw\phi_{xz} - 2vw\phi_{yz} = 0. \quad (14)$$

This is the steady, nonconservative full potential equation written in three-dimensional Cartesian coordinates. The velocity components ϕ_x , ϕ_y and ϕ_z have been replaced by u , v and w , respectively.

The unsteady version of the nonconservative full potential equation can be derived in a similar fashion and is given by

$$(a^2 - u^2)\phi_{xx} + (a^2 - v^2)\phi_{yy} + (a^2 - w^2)\phi_{zz} - 2uv\phi_{xy} - 2uw\phi_{xz} - 2vw\phi_{yz} = \phi_{tt} + 2u\phi_{xt} + 2v\phi_{yt} + 2w\phi_{zt}. \quad (15)$$

More on the characteristics of the full potential equation, both conservative and nonconservative forms, is presented in Section 2.10.

2.5. Transonic small disturbance (TSD) potential equation

Another potential equation formulation used for computing transonic flows about aircraft is the transonic small-disturbance (TSD) potential equation. Many of the numerical algorithm breakthroughs realized in solving the full potential equation were first developed using the simpler TSD potential formulation. The TSD potential equation is derived from the full potential equation by first defining a small-disturbance velocity potential φ

$$\nabla\varphi = \mathbf{q} - \mathbf{q}_\infty,$$

where \mathbf{q} is the usual local velocity vector and \mathbf{q}_∞ is the freestream velocity vector, which is assumed to be aligned with the x direction. It is defined by

$$\mathbf{q}_\infty = u_\infty \mathbf{i}.$$

With the above definitions, the small-disturbance velocity components of $\nabla\varphi$ are given by

$$\varphi_x = u - u_\infty, \quad \varphi_y = v, \quad \varphi_z = w.$$

Derivation of the TSD potential equation begins by substituting these small-disturbance velocity components into the full potential equation [either Eq. (14) for steady flows or Eq. (15) for unsteady flows]. Then after neglecting small terms according to the small-disturbance assumptions given by

$$\frac{\varphi_t}{u_\infty} \approx \frac{\varphi_x}{u_\infty} \approx \frac{\varphi_y}{u_\infty} \approx \frac{\varphi_z}{u_\infty} \ll 1,$$

$$\varphi_{tt} \approx \varphi_{xt} \approx \varphi_{yt} \approx \varphi_{zt} \approx \varphi_{xx} \approx \varphi_{yy} \approx \varphi_{zz} \approx 1,$$

the three-dimensional unsteady TSD potential equation becomes

$$\left[1 - M_\infty^2 - M_\infty^2 \left(\gamma + 1 \right) \frac{\varphi_x}{u_\infty} \right] \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = \frac{1}{a_\infty^2} (\varphi_{tt} + 2u_\infty \varphi_{xt}). \quad (16)$$

One term containing a small-disturbance quantity (the first term) survives the small-disturbance analysis. This is because for transonic flow

$$1 - M_x^2 \approx M_x^2(\gamma + 1) \frac{\varphi_x}{u_x}$$

Eq. (16) is valid for subsonic, supersonic and transonic flows that satisfy the original full potential equation assumptions (inviscid, isentropic and irrotational flow) and that are a “small disturbance” away from freestream. An essential ingredient of Eq. (16), and all transonic governing equations, is that they are nonlinear, i.e., the $\varphi_x \varphi_{xx}$ term is nonlinear. This term is required to predict shock waves, which are inherently a nonlinear phenomena. Linearization of a transonic flow governing equation removes the essential mathematics required to predict shock waves.

Another variation of Eq. (16) that has been used in many applications (see, e.g., [1,23]) is the low-frequency TSD equation. If the frequency of oscillation of the problem under consideration is small enough, then the φ_{tt} term can be neglected yielding

$$\left[1 - M_x^2 - M_x^2(\gamma + 1) \frac{\varphi_x}{u_x} \right] \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = \frac{2u_x}{a_x^2} \varphi_{xt}. \quad (17)$$

A convenient nondimensional measure of a problem’s frequency is obtained using the reduced frequency defined by $k = \omega c / u_x$, where ω is the problem’s physical oscillation frequency measured in cycles per second. When the reduced frequency of a specific unsteady problem is less than about 0.2 (and all other TSD assumptions are valid), then the low-frequency TSD equation is usually considered to be valid.

Of course, the steady TSD equation is obtained by neglecting all time derivatives in Eq. (16), which yields

$$\left[1 - M_x^2 - M_x^2(\gamma + 1) \frac{\varphi_x}{u_x} \right] \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = 0. \quad (18)$$

Eqs. (16)–(18) are in nonconservative form, i.e., not all variable coefficients are inside the outer differentiation. To transform these equations into conservative form is easily accomplished and yields [e.g., for Eq. (18)]

$$\left[(1 - M_x^2) \varphi_x - M_x^2(\gamma + 1) \frac{\varphi_x^2}{2u_x} \right]_x + \varphi_{yy} + \varphi_{zz} = 0. \quad (19)$$

Eqs. (16)–(19) represent classical forms of the TSD equation, but several other forms exist, including forms derived to better predict transonic flows on swept wings. For more information on other TSD equation forms see van der Vooren et al. [24] or Slooff [25].

Additional useful equations can be obtained from Eq. (16). For example, by neglecting the last small-distur-

bance term, the unsteady small-disturbance equation given by

$$(1 - M_x^2) \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = \frac{1}{a_x^2} (\varphi_{tt} + 2u_x \varphi_{xt}) \quad (20)$$

is obtained. This equation is linear and valid for either subsonic or supersonic flow, but not transonic flow. As in the TSD potential equation case, a steady version of this equation is obtained by neglecting all time terms, which yields

$$(1 - M_x^2) \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = 0. \quad (21)$$

This is the famous Prandtl–Glauert equation and can be used to describe steady, small-disturbance potential flow. Lastly, if the flow is assumed to be incompressible (i.e., $a_x \rightarrow \infty$, $M_x \rightarrow 0$) the small-disturbance potential equation, valid for either steady or unsteady flow, becomes

$$\varphi_{xx} + \varphi_{yy} + \varphi_{zz} = 0, \quad (22)$$

which is another version of Laplace’s equation. This version is based on the small-disturbance potential function and differs from Eq. (13) in the boundary conditions that are applied, both in the freestream and at the airfoil/wing surface.

2.6. Small-disturbance boundary conditions

The freestream boundary condition consistent with small-disturbance theory is that all disturbances must vanish in the freestream, i.e.,

$$\varphi_{x,\infty} = \varphi_{y,\infty} = \varphi_{z,\infty} = 0.$$

The flow tangency boundary condition for a typical “thin” airfoil or wing used in conjunction with any of the small disturbance formulations presented in the last section is generally derived as follows: The standard tangency condition can be implemented at the airfoil or wing surface using

$$\left(\frac{\varphi_z}{u} \right)_{ws}^{\pm} = \left(\frac{\varphi_z}{\varphi_x + u_x} \right)_{ws}^{\pm} = \frac{dg^{\pm}(x, y)}{dx},$$

where x is aligned with the freestream direction, y is in the span direction and z is in the vertical direction. The subscript “ws” indicates that the boundary condition is applied at the wing surface. The functions $g^+(x, y)$ and $g^-(x, y)$ define the upper and lower wing surfaces, respectively. The small disturbance version of this boundary condition is obtained by making two simplifications. First, φ_x is neglected relative to u_x in the middle term denominator. Second, the flow tangency boundary condition is applied at the airfoil slit, i.e., at $z = 0$, instead of the airfoil surface. This latter approximation greatly simplifies the volume grid generation process for TSD

potential applications because geometrical surfaces do not have to be fitted with grids. This is a key reason that this formulation was so widely used for three-dimensional transonic flow applications in the early years of CFD development. The final small-disturbance flow-tangency boundary condition with these simplifications becomes

$$\varphi_z(x, y, 0^\pm) = u_\infty \frac{dg^\pm(x, y)}{dx}.$$

This expression approximates the required flow tangency boundary condition at the airfoil surface to an accuracy consistent with small-disturbance theory. A key flaw in small-disturbance theory is displayed in this boundary condition at (for example) an airfoil leading edge where the slope of the surface becomes infinite, and accurate boundary condition implementation is impossible. This difficulty is a symptom of the breakdown in small-disturbance theory at stagnation points. The streamwise velocity component perturbation becomes large and is actually equal to the freestream velocity at the stagnation point. This fundamental limitation in the TSD potential equation approach is the primary reason that its use has declined in recent years.

2.7. Mathematical properties

The primary motivation for studying the nature of partial differential equations (PDEs) in the present context is to gain insight into the physics they describe and to develop guidelines for the implementation of numerical solution procedures. Different equation types generally require different solution algorithms. With this purpose in mind, consider the following general quasi-linear, second-order PDE:

$$Au_{xx} + Bu_{xy} + Cu_{yy} = F, \quad (23)$$

where u is an arbitrary dependent variable and A , B , C and F are (at most) functions of x , y , u , u_x and u_y . This equation can be studied and classified by considering the corresponding characteristic equation given by (for a derivation of the characteristic equation and additional discussion on this topic see Ames [26] and Mitchell [27])

$$A\left(\frac{dy}{dx}\right)^2 - B\left(\frac{dy}{dx}\right) + C = 0. \quad (24)$$

Using the quadratic formula, the two characteristic directions associated with Eq. (23) are given by

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{+B \pm \sqrt{B^2 - 4AC}}{2A}. \quad (25)$$

The nature of these characteristics determines the equation classification. Eq. (23) is hyperbolic if the character-

istics are real and distinct; i.e., if the discriminant of Eq. (25) 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 ($B^2 - 4AC < 0$).

By using the discriminant test described above, it can be shown that the TSD equation given by Eq. (18) (two-dimensional version) is hyperbolic when

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

and elliptic when

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

In other words, the sign of the first term coefficient determines the equation type. If the 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 potential equation are given by

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

Notice that these characteristic slopes are symmetric about the x -axis regardless of the local velocity vector orientation; i.e., the characteristics are not a function of the y component of the velocity φ_y . A sketch of the steady, two-dimensional TSD potential equation characteristics for a typical supersonic point is presented in Fig. 3. This situation, which is in dramatic contrast to the full potential or Euler formulations, where the characteristics are symmetric about the local stream direction, has certain simplifying implications regarding spatial discretization approximations for the TSD potential equation. In particular, it is much easier to construct a spatial discretization scheme for the TSD potential equation with a numerical domain of dependence that is compatible with the PDE's mathematical domain of dependence. As seen in Fig. 3, as v grows relative to u , the x -axis symmetry condition for the characteristics becomes more and more nonphysical. This is a direct result of the small-disturbance assumption that requires v to remain small in order to keep the TSD potential equation valid.

Classification of the steady full potential equation given by Eqs. (9) and (10) is difficult because this set of equations is not in the standard form given by Eq. (23). However, the nonconservative full potential equation [Eq. (14)] is ideally suited for this purpose. This equation written in two dimensions is given by

$$(a^2 - u^2)\phi_{xx} - 2uv\phi_{xy} + (a^2 - v^2)\phi_{yy} = 0. \quad (26)$$

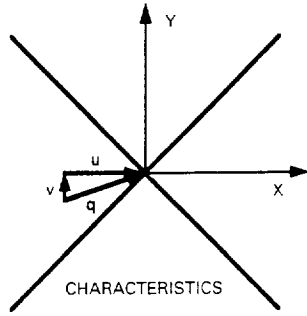


Fig. 3. Sketch of the two-dimensional TSD equation characteristics for a typical supersonic point.

where (again) u and v have been substituted for ϕ_x and ϕ_y , and ϕ is the full velocity potential. The discriminant defined in Eq. (25) for the full potential equation is given by

$$B^2 - 4AC = a^2(q^2 - a^2),$$

where $q^2 = u^2 + v^2$. It can be easily seen from the above equation that the full potential equation is hyperbolic for supersonic flow ($q > a$), parabolic for sonic flow ($q = 0$), and elliptic for subsonic flow ($q < a$). Even though this result is obtained for the nonconservative form of the full potential equation, it is also valid for the conservative form because both forms are mathematically equivalent.

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

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{-uv \pm \sqrt{a^2(q^2 - a^2)}}{a^2 - u^2}.$$

Notice that the characteristic directions are not symmetric about the x -axis as is the case with the TSD potential equation. Instead, the characteristics are symmetric about the stream direction. This can be shown by transforming the full potential equation [Eq. (26)] into a local stream and stream-normal coordinate system (s, n) using the following transformation

$$x = \frac{u}{q}s - \frac{v}{q}n, \quad y = \frac{v}{q}s + \frac{u}{q}n.$$

The resulting equation is given by

$$(a^2 - q^2)\phi_{ss} + a^2\phi_{nn} = 0, \quad (27)$$

where

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

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

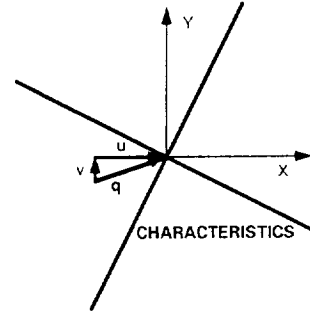


Fig. 4. Sketch of the two-dimensional full potential equation characteristics for a typical supersonic point. (a) Physical domain. (b) Computational domain.

Using the new version of the full potential equation given above, new characteristic directions can be derived and are given by

$$\left(\frac{ds}{dn}\right)_{1,2} = \frac{\pm a}{\sqrt{q^2 - a^2}}.$$

Of course, these characteristic slopes are real and distinct only for hyperbolic flow ($q > a$). Since they are equal in magnitude, but opposite in sign, they are symmetric about the stream direction. A sketch of the steady, two-dimensional full potential equation characteristics for a typical supersonic point is presented in Fig. 4. Note the comparison with the TSD characteristics presented in Fig. 3. The mathematical domain of dependence associated with the full potential equation follows the physical domain of dependence more generally than does the TSD potential equation. Spatial discretization schemes designed to solve the full potential equation must take this fact into account for proper numerical operation.

2.8. Transformation techniques

So far all potential equation formulations have been presented using Cartesian coordinates. Often, before solution algorithms can be implemented, the governing equations must be transformed from the physical domain (Cartesian coordinates) into some suitable computational domain. This is a requirement for finite-difference and some finite-volume methods, but not methods based on unstructured grid approaches, which are described in Section 3.10. Even applications that use Cartesian coordinates in the computational domain, e.g., most TSD applications, 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-condition

implementation and grid clustering at geometrical surfaces can be achieved without undue difficulty.

A general, independent variable transformation written for unsteady, three-dimensional applications, which maintains strong conservation-law form of the original governing equation, is given by (see [28-31] for more discussion on this type of transformation procedure)

$$\begin{aligned} \xi &= \xi(x, y, z, t), & \eta &= \eta(x, y, z, t), & \zeta &= \zeta(x, y, z, t), \\ \tau &= t, \end{aligned} \quad (28)$$

where x, y, z and t represent the Cartesian-coordinate physical domain and ξ, η, ζ and τ the computational domain (see Fig. 5 for a two-dimensional example). The conservative full potential equation written in Cartesian coordinates [Eqs. (4) and (8)] is transformed into the computational domain ξ, η, ζ, τ coordinate system by ap-

plying the standard chain rule written for the inverse of the transformation given by Eqs. (28), namely

$$\begin{aligned} \frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta}, \\ \frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta}, \\ \frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta}, \\ \frac{\partial}{\partial t} &= \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \zeta_t \frac{\partial}{\partial \zeta} + \frac{\partial}{\partial \tau}, \end{aligned} \quad (29)$$

where the terms containing derivatives of τ with respect to x, y or z from the first three lines in Eq. (29) are zero because of the dependence of τ on only t . However, the terms containing derivatives of ξ, η , or ζ with respect to t are, in general, not zero. The full potential equation given by Eqs. (4) and (8) transformed using Eqs. (28) becomes

$$\left(\frac{\rho}{J}\right)_\tau + \left(\frac{\rho U}{J}\right)_\xi + \left(\frac{\rho V}{J}\right)_\eta + \left(\frac{\rho W}{J}\right)_\zeta = 0, \quad (30a)$$

$$\rho = \left\{ 1 + \frac{\gamma-1}{2} [M_\infty^2 - 2\phi_\tau - (U + \xi_t)\phi_\xi - (V + \eta_t)\phi_\eta - (W + \zeta_t)\phi_\zeta] \right\}^{1/(\gamma-1)}, \quad (30b)$$

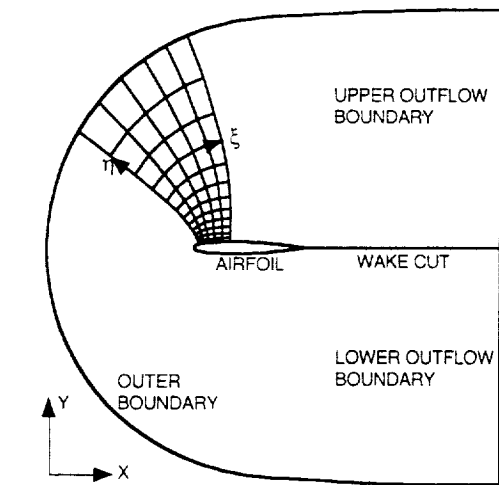
where

$$\begin{aligned} U &= \xi_t + A_1 \phi_\xi + A_4 \phi_\eta + A_5 \phi_\zeta, \\ V &= \eta_t + A_4 \phi_\xi + A_2 \phi_\eta + A_6 \phi_\zeta, \\ W &= \zeta_t + A_5 \phi_\xi + A_6 \phi_\eta + A_3 \phi_\zeta, \end{aligned} \quad (31a)$$

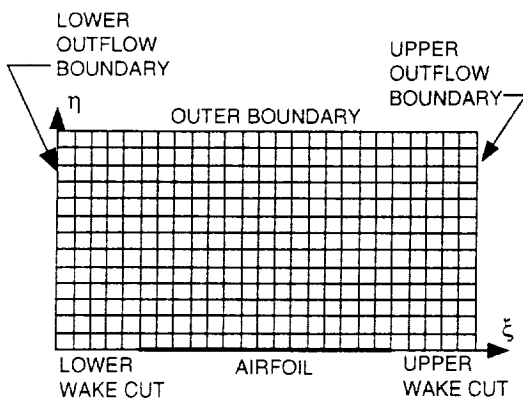
and

$$\begin{aligned} A_1 &= \nabla \xi \cdot \nabla \xi = \xi_x^2 + \xi_y^2 + \xi_z^2, \\ A_4 &= \nabla \xi \cdot \nabla \eta = \xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z, \\ A_2 &= \nabla \eta \cdot \nabla \eta = \eta_x^2 + \eta_y^2 + \eta_z^2, \\ A_5 &= \nabla \xi \cdot \nabla \zeta = \xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z, \\ A_3 &= \nabla \zeta \cdot \nabla \zeta = \zeta_x^2 + \zeta_y^2 + \zeta_z^2, \\ A_6 &= \nabla \eta \cdot \nabla \zeta = \eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z, \\ J &= \xi_x \eta_y \zeta_z + \xi_y \eta_z \zeta_x + \xi_z \eta_x \zeta_y - \xi_z \eta_y \zeta_x - \xi_y \eta_x \zeta_z \\ &\quad - \xi_x \eta_z \zeta_y \\ &= (x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta - x_\zeta y_\zeta z_\eta - x_\eta y_\xi z_\zeta \\ &\quad - x_\zeta y_\eta z_\xi)^{-1}. \end{aligned} \quad (31b)$$

In Eqs. (31a) and (31b), U, V and W are the contravariant velocity components along the ξ, η and ζ coordinate



(a)



(b)

Fig. 5. Numerically generated airfoil transformation $[(x, y) \leftrightarrow (\xi, \eta)]$ showing a "C" grid topology. (a) Transonic flow. (b) Supersonic flow.

directions, respectively; A_1-A_6 are metric quantities; and J is the determinant of the transformation Jacobian. The metric quantities given above in Eq. (31b) are evaluated using the following metric identities:

$$\begin{aligned} \xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta), & \eta_x &= J(y_\zeta z_\xi - y_\xi z_\zeta), \\ \xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta), & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi), \\ \xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta), & \eta_z &= J(x_\zeta y_\xi - x_\xi y_\zeta), \\ \xi_x &= J(y_\zeta z_\eta - y_\eta z_\zeta), & \zeta_x &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z, \\ \xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta), & \zeta_y &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z, \\ \xi_z &= J(x_\zeta y_\eta - x_\eta y_\zeta), & \zeta_z &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z. \end{aligned}$$

The transformed full potential governing equation given by Eqs. (30a) and (30b) can be used for general geometries in which the aerodynamic surface of interest is mapped to a constant coordinate line in the computational domain, even if the surface is moving in time. For steady flow problems, the proper transformed full potential governing equation is simply obtained by setting all time terms in the above equations equal to zero. With this mapping procedure, for either steady or unsteady problems, application of the flow-tangency boundary condition is easy and accurate to implement. For example, if the aerodynamic surface of interest is defined by $F(x, y, z, t) = 0$ then the flow tangency boundary condition is given by

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \mathbf{q} \cdot \nabla F = 0. \tag{32}$$

In the ξ, η, ζ, τ computational domain, assuming (for example) that the aerodynamic surface of interest F is mapped to an $\eta = \text{constant}$ surface, the flow-tangency boundary condition becomes

$$\begin{aligned} \eta_t + \mathbf{q} \cdot \nabla \eta &= \eta_t + (\phi_x \mathbf{i} + \phi_y \mathbf{j} + \phi_z \mathbf{k}) \cdot (\eta_x \mathbf{i} + \eta_y \mathbf{j} + \eta_z \mathbf{k}) \\ &= V = 0. \end{aligned} \tag{33}$$

More simply stated, the contravariant velocity component in the η -direction V must vanish at the $\eta = \text{constant}$

surface where flow tangency is required. For problems in which the boundary does not change with time, the proper flow tangency boundary condition is obtained from the above condition by simply setting the time term to zero. If the $\eta = \text{constant}$ surface is a $y = \text{constant}$ symmetry plane, the above flow tangency boundary condition is also generally applicable.

2.9. Shock wave capture criteria

The full potential equation formulations given above are valid for isentropic, irrotational flows about arbitrary shapes. To obtain physically realistic results, however, the full potential equation is restricted to shapes and to flows for which viscous effects (in particular, flow separation) are not important. The full potential equation is also restricted to flows that contain at most weak shock waves. Thus, allowable freestream conditions range from incompressible ($M_\infty \approx 0$) to supersonic ($M_\infty > 1$), providing the shock waves are "weak." The weak shock wave condition is approximately satisfied if the maximum normal shock Mach number never exceeds 1.3. Fig. 6 shows two typical flow situations which are generally valid for full potential equation simulations, Fig. 6a showing a typical transonic flow field and Fig. 6b showing a typical supersonic flow field.

The full potential formulation, despite the isentropic assumption, is approximately valid for these weak shock wave cases because the entropy produced by such weak shock waves is very small. This is evident by looking at the entropy change across a shock wave (Δs) as a function of the upstream normal Mach number component (M_n), which is given by [32]

$$\Delta s = s_2 - s_1 = O(M_n^2 - 1)^3,$$

where s_1 and s_2 are entropy values upstream and downstream of the shock wave, respectively. Note that for small values of $M_n^2 - 1$ the entropy production is very small, and the isentropic assumption is valid. A comparison of the isentropic shock jump relation written for

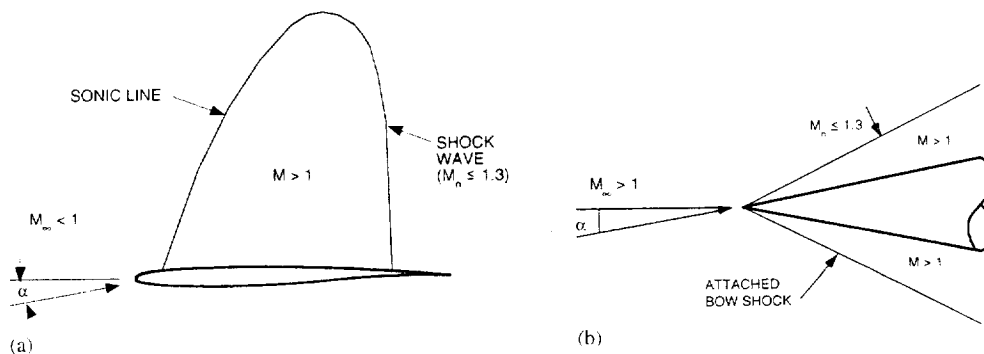


Fig. 6. Typical transonic and supersonic flow cases for which the full potential formulation is valid.

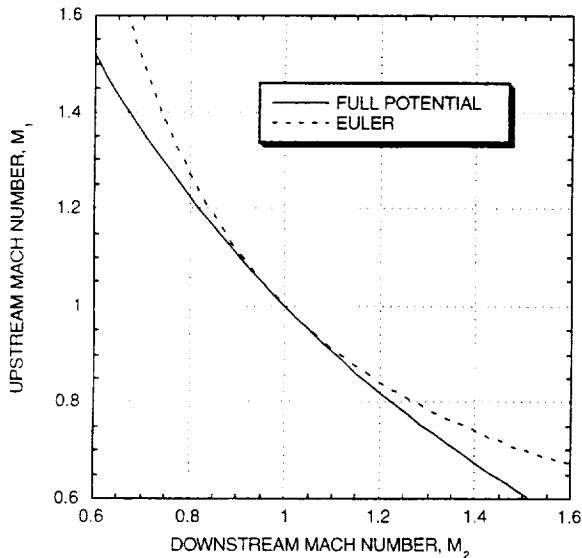


Fig. 7. A comparison of the full potential and Rankine-Hugoniot (Euler equations) shock-jump relations for a one-dimensional normal shock wave, taken from Steger and Baldwin [33].

a one-dimensional normal shock wave, given by

$$\left[\left(1 - \frac{\gamma-1}{\gamma+1} \phi_x^2 \right)^{1/(\gamma-1)} \phi_x \right]_1 = \left[\left(1 - \frac{\gamma-1}{\gamma+1} \phi_x^2 \right)^{1/(\gamma-1)} \phi_x \right]_2$$

with the Euler shock jump relations (the Rankine-Hugoniot conditions) is given in Fig. 7 (taken from Steger and Baldwin [33]). For a local upstream Mach number (M_1) at or below 1.3 a reasonable approximation is obtained by the isentropic formulation. Values of M_1 less than one correspond to expansion shock waves, which are physically excluded by entropy. Since potential formulations are isentropic, another mechanism must be introduced to exclude expansion shock waves. The introduction of dissipation or artificial viscosity by a suitable upwind-biased discretization scheme in supersonic regions of flow achieves this effect and will be discussed in more detail subsequently.

2.10. Conservative versus nonconservative forms

In 1954 Lax [34] showed the importance of using conservative form for the fluid dynamic equations when shock capturing schemes are used. This conclusion is based on the fact that nonconservative differences of discontinuous flow quantities across shock waves are numerically inaccurate. Differences of flow variables, which are conserved, that is, which are continuous across

shock waves, are much more accurate. This, of course, is a relative statement. Use of conservative form does not by itself guarantee an accurate resolution of captured shock waves. Many other numerical considerations play an important role. However, if a shock-capturing scheme is used to solve a nonconservative form of the governing flow equations, no matter what the numerical scheme characteristics, significant errors in the shock wave position and strength can result.

If this is true and was known as early as the mid 1950s, then why was there so much attention devoted to solving the nonconservative potential equation in the early 1970s? The answer to this question is twofold. First, the nonconservative forms of the TSD and full potential equations are more convenient to solve because of the sign change associated with the leading term coefficient at or near the sonic line. This allows a simple construction for type-dependent numerical schemes. This property is not shared by conservative forms of these equations. The second reason is that the error induced by nonconservative form for a shock-capturing computation involving only weak shocks is not large, and it fortuitously produces results (for inviscid computations) in better agreement with experiment than conservative schemes.

In the previous section a plot comparing the shock polars for the Euler and full potential equations is presented (see Fig. 7). Although a similar analytic shock polar for the nonconservative full potential equation cannot be derived, shock jumps obtained computationally can be compared (see Gregg and Henne [35] for a number of computations where this has been done). The nonconservative shock polar, thus computed, compares more favorably with experiment than with conservative results. This fact has caused the nonconservative potential formulation to be utilized in many different applications. But why does this behavior exist and is conservative form really the correct form to use?

The superior experimental correlation that nonconservative potential methods exhibit relative to conservative methods is due to an effective mass source introduced at shocks. This numerically generated “error” fortuitously models the reduced shock pressure rise caused by the shock/boundary-layer interaction, and therefore, in most cases, produces better agreement with the experimental pressure distribution than a conservative result. Newman and South [36,37] present a quantitative description of this behavior. In this study, conservative and nonconservative TSD potential solutions are computed about a 10%-thick, non-lifting, parabolic-arc airfoil. Pressure distributions and streamline deflection patterns for this problem at two different freestream Mach numbers ($M_\infty = 0.84$ and $M_\infty = 0.95$) are presented in Fig. 8.

The nonconservative shock wave at the lower freestream Mach number is weaker and slightly forward of the conservative shock. For the higher freestream Mach

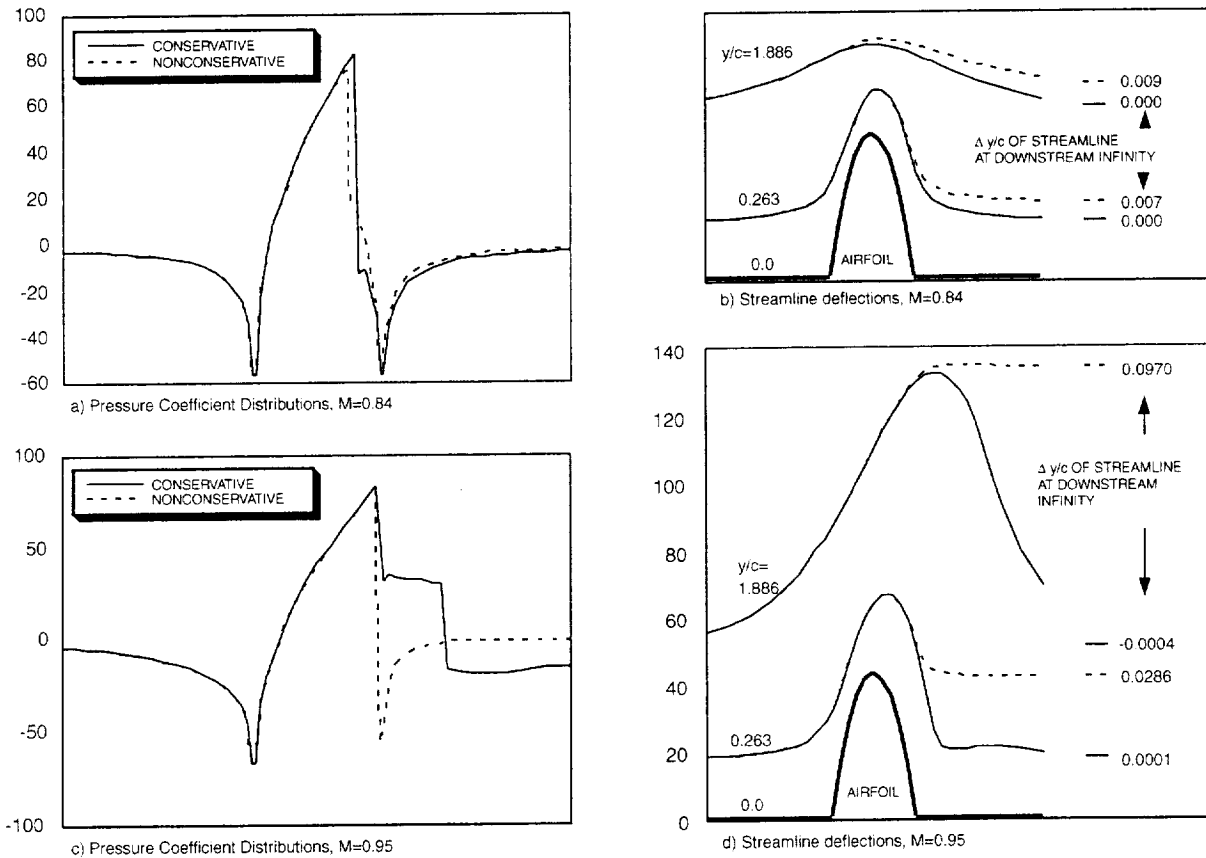


Fig. 8. Computed symmetry line pressure coefficients and streamline deflections for non-lifting transonic flow past a 10%-thick parabolic-arc airfoil (taken from Newman and South [36,37]).

number case, the conservative result exhibits a so-called “fish-tail” shock wave solution (an oblique shock emanating from the trailing edge followed by a normal shock about half a chord downstream of the airfoil trailing edge). The nonconservative result for the higher Mach number case is quite different than the conservative result exhibiting only a single normal shock at the airfoil trailing edge. The computed streamline deflection patterns in Fig. 8 show the cause for these pressure distribution discrepancies, which seem to increase with Mach number. Note that the vertical scale has been magnified by a factor of about 20 to accentuate the situation. The conservative streamlines entering the flow field are the same height as those leaving. However, the nonconservative streamlines are deflected upward (at the approximate position of the shock wave) indicating a numerical error resulting in effective mass addition at the shock wave. Thus, use of the nonconservative form destroys global mass conservation when captured shock waves are present.

Note: This situation is alarming for external flow calculations but disastrous for internal flow situations where a global mass balance is even more important.

When viscous corrections are added to the simulation, the conservative versus nonconservative controversy changes. The addition of viscous corrections to inviscid formulations is important for many calculations in the subsonic and transonic cruise regime. If the viscous correction procedure is accurate in simulating all aspects of viscous flow, then the nonconservative formulation will still produce mass sources at shock waves, and therefore, introduce errors into the solution. The conservative formulation, with accurate viscous corrections, will produce the correct physical answer, at least, within the limitations of the irrotational and isentropic assumptions. The ultimate formulation must be based on the mathematically sound conservative form. Most recently produced and utilized potential flow codes have been based on conservative formulations, and this trend is anticipated to continue.

2.11. Full potential equation nonuniqueness

In 1981 Steinhoff and Jameson [38] described a nonuniqueness problem for the steady, conservative full potential equation. Additional work further exploring this phenomena is presented in Salas et al. [39–41] for the steady full potential equation, in Dowell et al. [42] for the unsteady TSD equation, and Williams et al. [43] for the unsteady TSD equation run in both steady and unsteady modes with and without a simple boundary layer correction procedure. In these studies the nonuniqueness exists for both steady and unsteady flows. For steady flows, it manifests itself in the form of multiple, significantly different solutions at one angle of attack. These multiple solutions exhibit dramatically different values of circulation, and therefore, dramatically different values of lift. For unsteady flows it manifests itself in the form of non-zero mean lift for harmonic pitch oscillations of a symmetric airfoil at a zero mean angle of attack. For both steady and unsteady computations, the number of iterations or time-steps required for the nonuniqueness to appear is large, being an order of magnitude more than the number of iterations required for tight convergence of unique computations.

The nonuniqueness has only been exhibited in two-dimensional simulations, i.e., airfoil computations, that utilize the conservative form of the full potential or TSD equations. For a particular airfoil, the nonuniqueness occurs over a narrow freestream Mach number range involving transonic flow conditions, and thus, involves a shock wave on at least the upper or lower airfoil surface. This anomaly has not been demonstrated for the nonconservative full potential equation (see Salas et al. [41]) nor for the unsteady full potential equation (see Murthy [44]). It also has not been demonstrated for any three-dimensional potential formulation involving traditional aerodynamic problems, e.g., transport wing or wing-body computations. However, it can be demonstrated in three dimensions for wing applications when the aspect ratio is set to a large value, e.g., at or above 24 (see Holst [45]). For such computations, in the appropriate transonic Mach number range, the nonunique solution exists at the wing root, where the solution is essentially two dimensional in nature, but transitions to a unique solution at the wing tip. Another interesting characteristic is that a nonisentropic correction to the two-dimensional conservative full potential equation (as described in Section 3.3) restores a unique lift-angle-of-attack relationship, at least for the cases presented in Zi-qiang and Xue-Song [46]. Finally, McGrattan [47] demonstrates that for transonic airfoil solutions with very weak shocks involving a 3%-thick airfoil, the conservative full potential and the Euler equations both produce nonunique solutions, i.e., a significant nonzero value of lift for a symmetric airfoil at zero angle of attack. The two solutions are not identical, but are very close.

This suggests the nonuniqueness difficulty, as speculated in earlier studies, is not due to the isentropic, irrotational nature of potential formulations, but that the cause lies elsewhere, perhaps in how the Kutta condition is implemented.

More quantitative characteristics of this behavior are exhibited in Fig. 9, where several lift versus angle-of-attack plots are presented. The first three curves in this figure; CFP (2D), conservative full potential in two dimensions; Euler (2D); and NFP (2D), nonconservative full potential in two dimensions; are from Salas et al. [41]. The last curve, CFP (3D), is a three-dimensional, conservative full potential result from Holst [45]. The first three curves have been computed using two-dimensional algorithms for the flow around a NACA 0012 airfoil at a freestream Mach number of 0.83. Because of the multi-valued nature of the CFP (2D) curve, its computation was achieved by specifying lift and computing the angle of attack as described in Salas et al. [41]. In each of these cases, the solution is obtained using a fine grid with a tight convergence criteria, i.e., numerical errors have been minimized. The last curve presented in Fig. 9 is computed using a three-dimensional algorithm for the flow about an isolated-wing with NACA 0012 airfoil sections at zero sweep, a taper ratio of 1.0 and an aspect ratio of 8.0. Thus, the root station of this simulation approximately matches the other two-dimensional simulations displayed in Fig. 9.

From Fig. 9, the following observations can be made: (1) The two-dimensional conservative full potential curve is nonunique, exhibiting three solutions for each angle of attack in the approximate range $-0.3^\circ < \alpha < 0.3^\circ$. The lift-curve slope for this result is nonphysical, exhibiting the wrong sign in the anomalous angle of attack range.

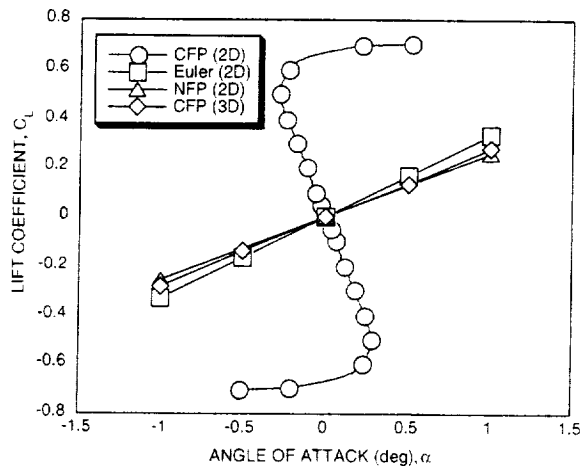


Fig. 9. Lift curves obtained from several different CFD codes for the NACA 0012 airfoil at a freestream Mach number of 0.83. The first three curves in this plot have been taken from Salas et al. [41] and the fourth curve [CFP (3D)] is from Holst [45].

(2) The Euler and nonconservative full potential curves exhibit unique results, i.e., one value of lift for one angle of attack. (3) The three-dimensional conservative full potential curve also exhibits unique results. The above observations describe the major known steady-flow characteristics of the full potential nonuniqueness problem. Explaining the cause is more difficult. It appears to be formulational in nature and not due to any aspect of numerical implementation, but a more definitive explanation is not available as of this writing. A primary ameliorating factor is that the nonuniqueness does not appear in traditional three-dimensional aerodynamic problems, where virtually all potential applications are implemented.

3. Algorithms and applications

3.1. Early TSD potential equation algorithms and applications

The first computations with a transonic potential formulation for an aerodynamic application involve the transonic small-disturbance (TSD) potential equation. As described in the previous chapter, the TSD potential formulation has the additional assumptions beyond the full potential formulation that the flow be aligned with a particular coordinate direction (usually the x -coordinate) and that only small changes or disturbances in the velocity components exist throughout the entire flow domain. In addition, flow tangency boundary conditions are applied along simplified surfaces that only approximate the geometry of interest, e.g., along the chord line of an airfoil instead of the actual airfoil surface. These additional assumptions greatly simplify implementation of a TSD potential equation solver, but are severely tested at stagnation points.

A key breakthrough in the field of CFD was the discovery of “type-dependent differencing” in 1971 by Murman and Cole [48] and was demonstrated using the TSD potential formulation for transonic airfoil simulations. Prior to 1971, simulation of transonic flow using a potential formulation was not possible. Inconsistencies between subsonic flow regions, which require central-differencing, and supersonic flow regions, which require upwind-differencing, caused numerical difficulties. The Murman–Cole algorithm “switches” the differencing type from central to upwind or vice versa as dictated by the local Mach number, maintaining stable operation for transonic flows, even those with strong shocks. This idea was extended to axisymmetric bodies by Bailey [49] and Krupp and Murman [50] and to three-dimensional isolated-wing applications by Bailey and Steger [51], Ballhaus and Bailey [52] and Newman and Klunker [53]. In all these applications the nonconservative form of the TSD equation is used.

The conservative form of the TSD equation is solved in two dimensions by Murman [54] and in three dimensions for isolated wings by Bailey and Ballhaus [55]. The type-dependent spatial difference scheme used to solve the two-dimensional TSD potential equation in conservative form can be presented by considering

$$f_x + g_y = 0,$$

which is a reformulation of the conservative TSD potential equation where

$$f = (1 - M_\infty^2)\phi_x - \frac{1}{2}M_\infty^2(\gamma + 1)\phi_x^2, \quad g = \phi_y.$$

The f and g quantities represent mass fluxes (or more appropriately, perturbations to the freestream mass fluxes) in the x and y directions, respectively. Note that all variables are inside the outer differentiation, a standard characteristic of conservation form. A discretization scheme valid for both subsonic and supersonic flow regions is given by

$$\frac{1}{\Delta x}(\bar{f}_{i+1/2,j} - \bar{f}_{i-1/2,j}) + \frac{1}{\Delta y}(g_{i,j+1/2} - g_{i,j-1/2}) = 0,$$

where the i and j subscripts indicate location in the finite-difference grid, such that $x = i\Delta x$ and $y = j\Delta y$, and \bar{f} is a modified flux defined by

$$\bar{f}_{i+1/2,j} = \mu_i f_{i+1/2,j} + (1 - \mu_i) f_{i-1/2,j}.$$

In the above equation, μ_i is a switching function defined by

$$\mu_i = \begin{cases} 1, & M_{i,j} \leq 1, \\ 0, & M_{i,j} > 1, \end{cases}$$

where $M_{i,j}$ is the local Mach number computed at point i, j . The above differencing scheme contains four different schemes or operators: (1) subsonic operator ($\mu_i = 1, \mu_{i-1} = 1$), (2) supersonic operator ($\mu_i = 0, \mu_{i-1} = 0$), (3) sonic-point operator ($\mu_i = 0, \mu_{i-1} = 1$), and (4) shock-point operator ($\mu_i = 1, \mu_{i-1} = 0$). This scheme is automatically second-order accurate and centrally differenced in all subsonic regions of flow and first-order accurate and upwind-differenced in all regions of supersonic flow. This scheme is also conservative, i.e., the fluxes in each cell have an identical flux in the immediately adjacent neighboring cell, such that all internal fluxes cancel identically.

A typical example (taken from Bailey and Ballhaus [55]) showing both TSD nonconservative and conservative inviscid pressure distributions compared with experiment is displayed in Fig. 10. These results are for a swept, isolated-wing configuration (ONERA M6) at rather harsh transonic flow conditions, $M_\infty = 0.92$, $\alpha = 3^\circ$. Note that the nonconservative results are in better agreement with experiment than the conservative results. However, both results still suffer serious disagreements. In particular, both computational results fail to predict

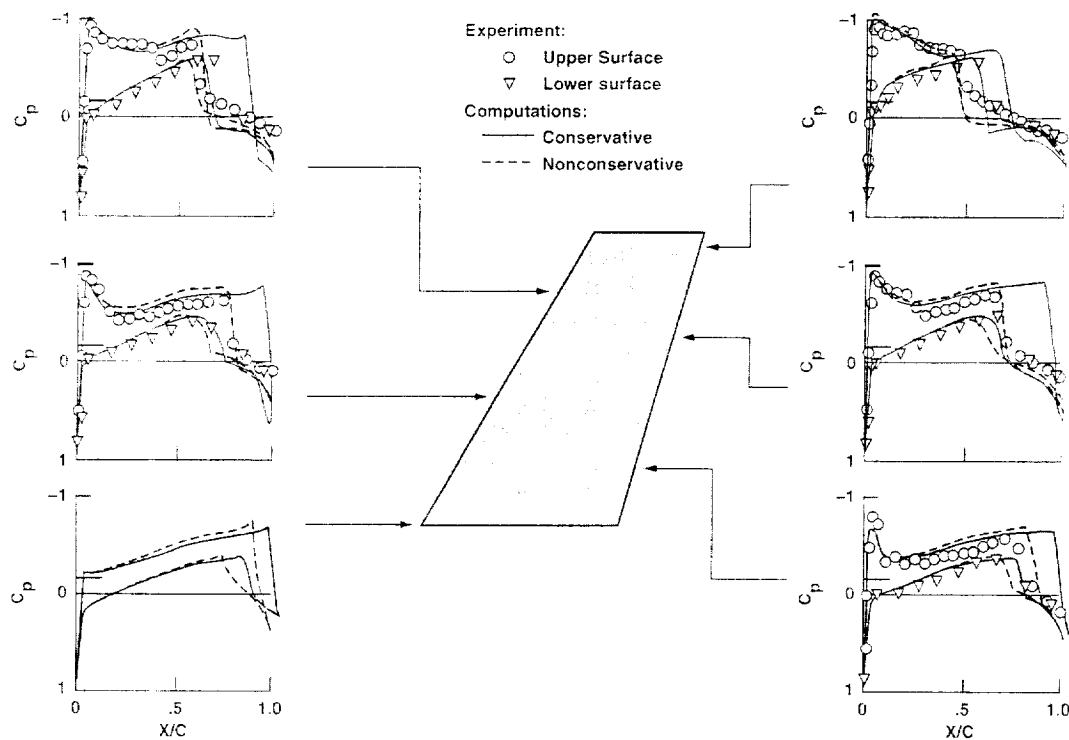


Fig. 10. Comparison of computed and experimental pressure coefficient distributions for an ONERA M6 wing, $M_\infty = 0.92$, $\alpha = 3^\circ$, taken from Bailey and Ballhaus [55].

the forward swept shock that is evident in the three outboard stations of the experiment. The large disagreement in the upper-surface aft-shock location for the conservative result is primarily caused by strong-shock boundary-layer interactions, which are not modeled in this inviscid computation. The conservative method does a better job of predicting the weaker shock wave on the lower wing surface. This result is largely interesting because it shows an early level of achievement in modeling transonic flow using the three-dimensional TSD equation.

Numerous efforts utilizing the TSD potential equation for simulating aerodynamic flows about a large variety of three-dimensional configurations have been reported in the literature. Examples of some of these efforts include Rohlfis and Vanimo [56], Schmidt and Hedman [57] and van der Vooren et al. [58,24] for wing and wing-body applications; Mason et al. [59] for wing and wing-body applications with viscous corrections; Albone et al. [60] and Firman [61] for wing, wing-body and wing-body-multiple-store computations with and without viscous corrections; Rae [62] and Rae and Lordi [63] for three-dimensional cascade computations; Shankar and Malmuth [64] for wing-body-canard computations; Phillips and Waggoner [65] for wing computations mounted inside wind-tunnel walls; and Boppe [66] and Boppe and

Stern [67] for wing, wing-body and wing-body-store-winglet computations. Most of these efforts utilize the nonconservative form of the TSD potential equation due to the fortuitously improved agreement with experimental pressure distributions. In some cases the conservative form of the TSD equation is available as an option as coding differences between these two TSD potential governing equation forms is not that great [compare Eqs. (18) and (19)].

Most of the TSD potential applications just listed utilize a sheared-stretched Cartesian-like mapping procedure, whereby the wing planform is mapped to a rectangle (see Fig. 11). Thus, each span station has the same number of grid points along the chord. The wing leading edge is positioned to lie between two grid lines. Thus, the infinite slope problem at the leading edge of a blunt wing does not cause any difficulties in applying the tangency boundary condition. The invalidity of the small disturbance assumption near the leading edge stagnation line is not changed by this grid placement strategy, but (at least) the slope is never actually infinite. In general, poor results are obtained from any TSD formulation in the vicinity of a blunt leading edge stagnation point line.

Several of the TSD methods listed above utilize the method of grid embedding (first introduced by Boppe [66]). This approach uses a coarse grid to cover

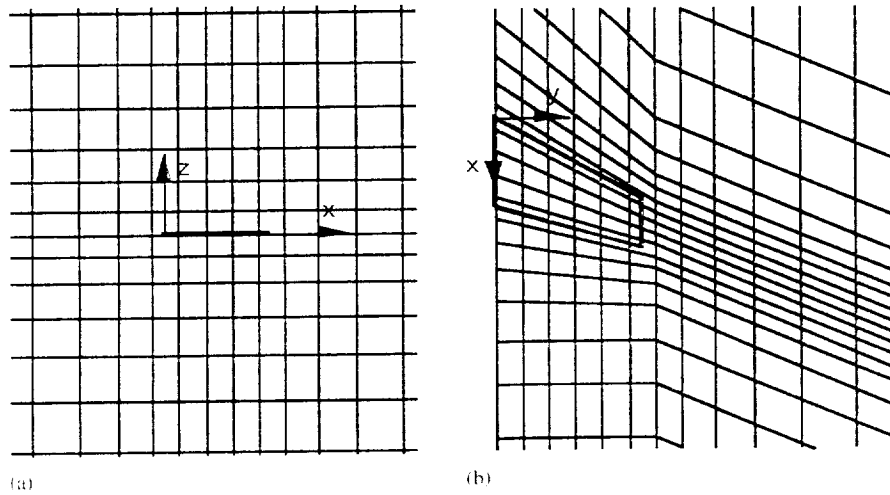


Fig. 11. Schematic of a typical TSD potential grid for an isolated-wing geometry showing shearing and stretching. (a) Wing cross-sectional view. (b) Wing planform view.

the entire flow field from far-field boundary to wing surface and a localized fine grid to resolve detail near the wing surface. Information from the outer coarse grid is passed to the fine inner grid outer boundary via interpolation. When the interface between the fine inner grid and the coarse outer grid is placed sufficiently far from the wing surface, no deterioration in solution accuracy at the wing surface can be detected. This concept is very attractive because it dramatically reduces the total number of grid points and the total amount of computer time required to achieve a given level of accuracy at the wing surface.

In all of the TSD references presented above the iteration scheme utilized is successive line overrelaxation (SLOR), which was a very popular potential equation relaxation scheme for the 1970s and early 1980s. A typical SLOR method for solving Laplace's equation [see Eq. (13)] is given by

$$\frac{1}{\Delta x^2} \left(\phi_{i-1,j}^n - 2\overline{\phi_{i,j}^{n+1}} + \phi_{i+1,j}^{n+1} \right) + \frac{1}{\Delta y^2} \left(\overline{\phi_{i,j}^{n+1}} - 2\overline{\phi_{i,j}^{n+1}} + \overline{\phi_{i,j}^{n+1}} \right) = 0, \quad (34)$$

where the i and j subscripts denote position in the finite-difference grid and the n superscript denotes iteration number. The $n+1$ superscript is an intermediate iteration level used to obtain the $n+1$ level by

$$\overline{\phi_{i,j}^{n+1}} = \omega \overline{\phi_{i,j}^{n+1}} + (1 - \omega) \phi_{i,j}^n.$$

In the above equation the parameter ω is a "relaxation factor". To obtain all the $n+1$ values for each $i = \text{constant}$ line requires the inversion of a scalar tridiagonal matrix. Because the grid points along the $i-1 = \text{con}$

stant line are updated prior to the $i = \text{constant}$ line (assuming the iteration scheme starts at $i = 1$ and proceeds to the maximum value of i), the values of ϕ at $i-1, j$ have already been updated, and thus, the superscript on $\phi_{i-1,j}$ is $n+1$. It should be noted that the above SLOR scheme is valid for both the small-disturbance potential and the full potential forms of Laplace's equation, with the only differences entering through the boundary conditions.

A standard von Neumann stability analysis of the above scheme shows that ω must be bounded by 0 and 2 for a stable iteration to result. Values of ω approaching 2 generally produce the fastest convergence with $\omega = 2$ being optimum as the number of grid points becomes infinite (see Garabedian [68] or Ames [26] for more information on this point). When $\omega > 1$ the above scheme is said to be overrelaxed, and when $\omega < 1$ the scheme is said to be underrelaxed. When $\omega = 1$ the above scheme becomes the traditional line Gauss-Seidel relaxation scheme.

The use of "overrelaxation" in relaxation schemes is a very important development that greatly improves convergence efficiency. It can be estimated (see Ames [26]) that the number of iterations N_{iter} required to drop the error by N_e orders of magnitude for the line Gauss-Seidel scheme is

$$N_{\text{iter}} \sim \frac{N_e}{2\Delta^2}, \quad (35)$$

and for the "optimally overrelaxed" SLOR scheme is

$$N_{\text{iter}} \sim \frac{N_e}{2\sqrt{2}\Delta}. \quad (36)$$

where Δ is the grid spacing used in the computation. It is easy to see from these estimates that the SLOR scheme can be one to two orders of magnitude faster than line Gauss–Seidel for problems utilizing fine grids. Although these theoretical convergence estimates are only (strictly speaking) valid for Laplace’s equation, qualitatively similar trends for convergence efficiency are exhibited for nonlinear potential formulations. Despite the significant enhancement in convergence efficiency offered by over-relaxation, other methods are still significantly faster, as will be seen shortly.

3.2. Early full potential equation algorithms and applications

The first nonlinear full potential algorithms were developed by Steger and Lomax [69] and Garabedian and Korn [70] and were utilized to simulate transonic flow about airfoils. The latter approach uses conformal mapping to map the domain external to the airfoil to a unit circle. This provides an elegant grid generation option and allows for simplified application of boundary conditions. Both of these approaches utilize the nonconservative form of the full potential equation and thus, produce shocked flows in which the position of the shock is affected by the numerical scheme. Nevertheless, these approaches, primarily the Garabedian–Korn algorithm, have often been used for many types of applications. Examples using the Garabedian–Korn code, including applications with a boundary layer correction procedure, are presented in Bauer et al. [71].

Throughout the 1970s numerous additional development efforts produced many two-dimensional and axisymmetric codes designed to solve the nonconservative full potential equation for transonic flows. For example, Carlson [72] produced a transonic airfoil analysis code that also included an inverse design option. South and Jameson [73] produced the first axisymmetric full potential flow solver called RAXBOD (see Keller and South [74] for additional details). This methodology is capable of simulating flows over sharp or blunt axisymmetric bodies at transonic and low supersonic speeds. Ives and Liutermoza [75,76] produced a transonic cascade analysis procedure with or without boundary layer correction. This two-dimensional approach has a stream-tube contraction correction that provides an approximation for three-dimensional effects. A number of codes have been produced for axisymmetric transonic inlets including the work of Arlinger [77], Baker [78], Caughey and Jameson [79] and Reyhner [80]. The iteration scheme in the majority of the above efforts is SLOR or a variety of SLOR.

The first three-dimensional transonic full potential solver (called FLO22), which also utilizes the nonconservative form of the full potential equation, was developed by Jameson [81]. This code utilizes the SLOR iteration scheme and a sheared-parabolic conformal-based map-

ping that effectively unwraps an isolated-wing into a rectangular domain. A typical FLO22 grid displayed in the physical domain is shown in Fig. 12. A key aspect of the FLO22 algorithm is the concept of “rotated differencing”. In this approach, the full potential equation is transformed into stream and stream-normal coordinates [see Eq. (27)]. The terms contributing to ϕ_{nn} are always centrally differenced. The terms contributing to ϕ_{ss} are centrally differenced in subsonic regions and upwind differenced in supersonic regions. With this approach, the computational domain of dependence always includes the physical domain of dependence, thus insuring improved stability for any orientation of the velocity vector. Another important contribution of this work is the description of the relaxation scheme as an iteration in “time”, i.e., a nonphysical time-like coordinate that behaves like the hyperbolic physical time coordinate. An important result of this time-like analysis is that temporal damping terms [of the form $\phi_{st} \approx (u/q)\phi_{xt} + (v/q)\phi_{yt}$] are required in supersonic regions of flow for a stable iteration scheme. For more information about this algorithm see Jameson [81] and Jameson et al. [82] or for information about FLO22 with a simple viscous correction procedure see Newman et al. [83].

A typical FLO22 surface pressure distribution compared with experiment (taken from Henne and Hicks [84]) is shown in Fig. 13. The experimental results displayed in this figure are for a supercritical wing low-mounted on a fuselage. The computational results are for only the wing portion of this geometry. A viscous correction is added using a two-dimensional strip approach. The flow conditions for this simulation include $M_\infty = 0.8$ and $\alpha = 2^\circ$. Agreement between the computation and experiment is generally good at all wing span stations. Note in particular that the double shock character for the upper-wing-surface solution is evident in both the experimental and computational results.

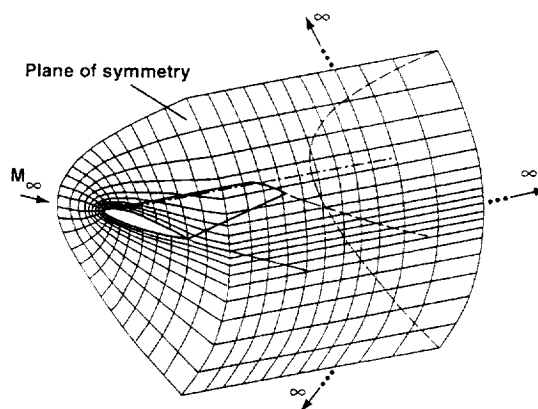


Fig. 12. Typical three-dimensional grid arrangement used in the FLO22 code involving a sheared-parabolic conformal-based mapping procedure, taken from Henne and Hicks [84].

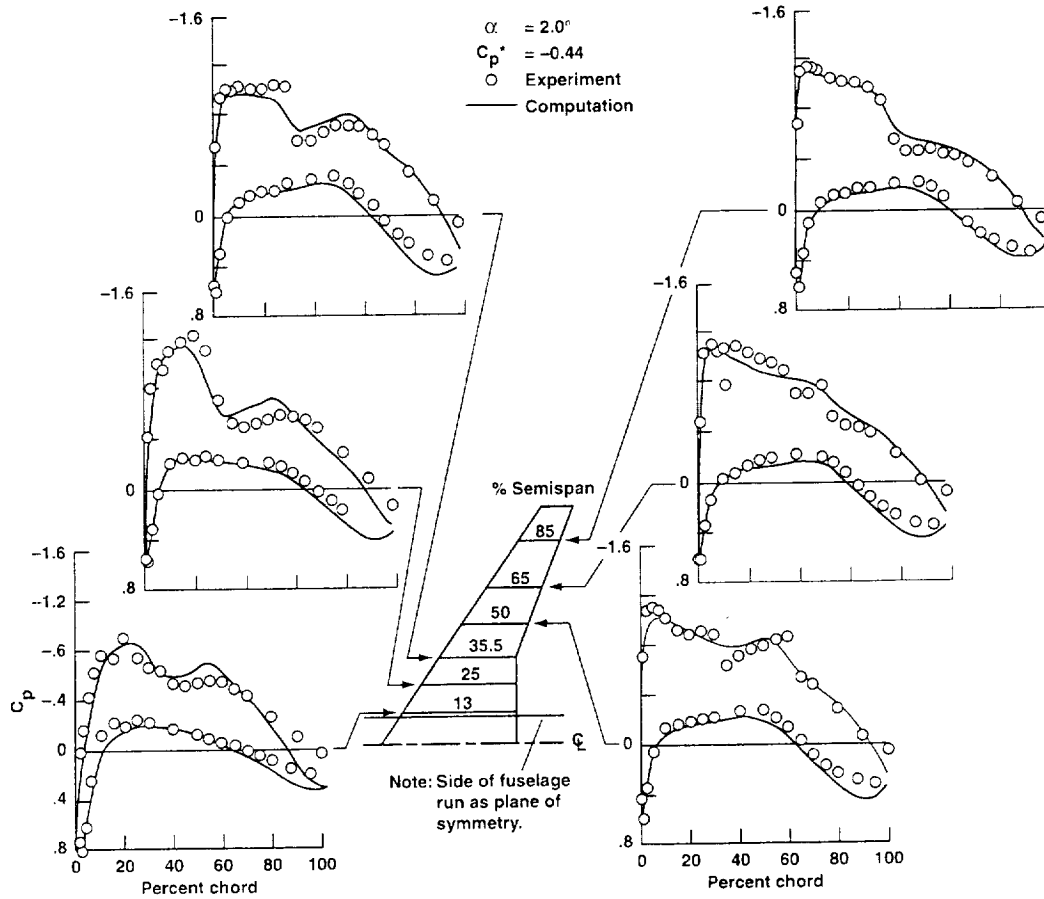


Fig. 13. Pressure coefficient comparisons for a transonic wing computation using the nonconservative full potential FLO22 computer code. $M_\infty = 0.8$, $\alpha = 2^\circ$, taken from Henne and Hicks [84].

Additional three-dimensional full potential applications based on nonconservative form include the inlet work of Reyhner [85], the wing simulations of Forsey and Carr [86], the cranked-wing applications of Chang and Tauber [87], and the quasi-unsteady efforts of Chang [88], Chang and Tung [89], Arieli et al. [90] and Egolf and Sparks [91]. The latter four efforts are focused on helicopter rotor blade applications involving either hover or advancing blade transonic flow simulations.

3.3. Conservative full potential equation algorithms

The primary goal of this section is to present spatial discretization schemes for solving the conservative full potential equation. The main emphasis is on finite-difference and finite-volume schemes. Finite-element schemes will be discussed in Section 3.11.

3.3.1. Finite-volume schemes

The first full potential solution method using conservative form was developed by Jameson [92] for solving

transonic airfoil flows in 1975. This work was followed closely by a series of very successful three-dimensional full potential solvers called FLO27, FLO28, and FLO30 [93–95]. The FLO27 code is capable of solving transonic flows about isolated wings or wings mounted on infinite circular cylinders. The FLO28 and FLO30 codes are very closely related derivatives of FLO27, but have more sophisticated abilities for treating the fuselage. A comparative study of these FLO codes can be found in Verhoff and O'Neil [96], where, in particular, the various fuselage modeling capabilities are evaluated. The actual grid mapping transformation used for most FLO27 and FLO28 wing calculations is very similar to the transformation used for the FLO22 code (see Fig. 12). The transformation used in the FLO30 code is somewhat more sophisticated and is described in detail in Caughey and Jameson [95]. All of the above FLO codes utilize the SLOR iteration scheme. Two examples where FLO30 has been upgraded with the addition of boundary layer corrections are described in Street [97] and Woodson et al. [98]. An example where the grid generation

capability used in FLO28 has been upgraded using the general grid generation capability of Thompson et al. [99] is described in Yu [100]. In the latter effort the improved grid generality is utilized to produce a transonic flow solution about a wing-body-nacelle-strut configuration. An example where the FLO27 code has been significantly enhanced is the work of Chen et al. [101]. In this effort a boundary layer correction code and an improved two-zone grid generation capability are utilized to compute transonic flow about a fuselage-pylon-nacelle configuration including both power-on and power-off cases.

The finite-volume spatial discretization scheme used in the FLO-series of codes (presented now in more detail) is designed to approximate the three-dimensional conservative full potential equation written in general non-orthogonal coordinates. This scheme, presented in two dimensions for brevity, is designed to solve

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

$$\rho = \left[1 + \frac{\gamma - 1}{2}(M_{\xi}^2 - U\phi_{\xi} - V\phi_{\eta})\right]^{1/(\gamma - 1)}, \quad (37)$$

where the above equations use ρ_{ξ} and a_{ξ} nondimensionalization. They are easily obtained from Eqs. (30a) and (30b) by setting all derivatives with respect to time and the ξ -coordinate direction to zero. Definitions for the contravariant velocity components (U and V) as well as all metric quantities can be obtained from Eqs. (31a) and (31b) by making similar simplifications. The spatial discretization scheme is given by

$$\bar{\delta}_{\xi} f_{i+1/2,j} + \bar{\delta}_{\eta} g_{i,j-1/2} = 0,$$

where δ_{ξ} and δ_{η} are backward, first-difference operators defined by

$$\bar{\delta}_{\xi}(\cdot)_{i,j} \equiv (\cdot)_{i,j} - (\cdot)_{i-1,j}, \quad \bar{\delta}_{\eta}(\cdot)_{i,j} \equiv (\cdot)_{i,j} - (\cdot)_{i,j-1}, \quad (38)$$

and the f and g fluxes are defined by

$$f_{i+1/2,j} = \frac{1}{2} \left[\left(\frac{\rho U}{J}\right)_{i+1/2,j+1/2} + \left(\frac{\rho U}{J}\right)_{i+1/2,j-1/2} \right],$$

$$g_{i,j+1/2} = \frac{1}{2} \left[\left(\frac{\rho V}{J}\right)_{i+1/2,j+1/2} + \left(\frac{\rho V}{J}\right)_{i-1/2,j+1/2} \right].$$

Computation of the above individual fluxes requires values of the density, the contravariant velocity components (U and V), and the determinant of the transformation Jacobian J , which, in turn, require derivatives of x , y and ϕ with respect to ξ and η . These computations are all performed at cell centers, i.e., at $i + 1/2, j + 1/2$, by using (for example)

$$\phi_{\xi} \Big|_{i+1/2,j+1/2} = \frac{1}{2}(\phi_{i+1,j+1} - \phi_{i,j+1} + \phi_{i+1,j} - \phi_{i,j}).$$

The above spatial discretization 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 a suitable recoupling term A of the form

$$A_{i,j} = -\frac{\varepsilon}{2} \bar{\delta}_{\xi} \bar{\delta}_{\eta} \left[\frac{\rho}{J} \left(A_1 - \frac{U^2}{a^2} + A_2 - \frac{V^2}{a^2} \right) \right]_{i+1/2,j+1/2}$$

$$\bar{\delta}_{\xi} \bar{\delta}_{\eta} \phi_{i,j},$$

where the parameter ε is a constant usually set equal to $\frac{1}{2}$ and $\bar{\delta}_{\xi}$ and $\bar{\delta}_{\eta}$ are forward, first-difference operators defined similarly to the backward operators in Eq. (38). The resulting scheme becomes

$$\bar{\delta}_{\xi} f_{i+1/2,j} + \bar{\delta}_{\eta} g_{i,j+1/2} + A_{i,j} = 0.$$

Addition of this new term recouples the odd and even points and represents a suitable spatial differencing scheme for subsonic regions of flow.

This scheme is stabilized in supersonic regions of flow by the explicit addition of artificial viscosity terms given by

$$\bar{P}_{i,j} = \frac{\nu \rho}{J a^2} (U^2 \delta_{\xi\xi} + UV \delta_{\xi} \delta_{\eta}) \phi_{i,j},$$

$$\bar{Q}_{i,j} = \frac{\nu \rho}{J a^2} (UV \delta_{\xi} \delta_{\eta} + V^2 \delta_{\eta\eta}) \phi_{i,j},$$

where the switching function ν is defined by

$$\nu = \max\left(0, 1 - \frac{M_c^2}{M^2}\right),$$

and the operators δ_{ξ} , δ_{η} , $\delta_{\xi\xi}$, and $\delta_{\eta\eta}$ are first and second central-difference operators defined by (for example)

$$\delta_{\xi}(\cdot)_{i,j} \equiv \frac{1}{2}[(\cdot)_{i+1,j} - (\cdot)_{i-1,j}],$$

$$\delta_{\xi\xi}(\cdot)_{i,j} \equiv (\cdot)_{i+1,j} - 2(\cdot)_{i,j} + (\cdot)_{i-1,j}.$$

The M_c parameter used above is a user-specified critical Mach number, defined in such a way that the spatial differencing scheme uses the subsonic scheme for values of local Mach number below M_c and the supersonic scheme for values of the local Mach number above M_c . In other words, the transition from central to upwind does not necessarily take place at the sonic line. Note that $M_c \leq 1$ for stability.

The final spatial differencing scheme is given by

$$\bar{\delta}_{\xi}(f_{i+1/2,j} + P_{i+1/2,j}) + \bar{\delta}_{\eta}(g_{i,j+1/2} + Q_{i,j+1/2})$$

$$+ A_{i,j} = 0, \quad (39)$$

where the P and Q terms are defined by

$$P_{i+1/2,j} = \begin{cases} \hat{P}_{i,j}, & U_{i+1/2,j} > 0, \\ -\hat{P}_{i+1,j}, & U_{i+1/2,j} < 0, \end{cases}$$

$$Q_{i,j+1/2} = \begin{cases} \hat{Q}_{i,j}, & V_{i,j+1/2} > 0, \\ -\hat{Q}_{i,j+1}, & V_{i,j+1/2} < 0. \end{cases} \quad (40)$$

A typical result computed with the above formulation is compared with experiment in Fig. 14. The geometry used in this comparison involves a wing/fuselage configuration with a supercritical wing. The wing quarter-chord sweep is 25° , the wing aspect ratio is 8.0 and the thickness-to-chord ratio is 0.12. The numerical results have been computed using the FLO30 computer code modified with an integral boundary layer correction procedure, which models the wing boundary layer thickness, the wake displacement thickness and the wake curvature (see Street [97]). The flow conditions for this case are $M_\infty = 0.819$, $\alpha = 1.96^\circ$, and $Re_c = 6 \times 10^6$, where \bar{c} is the mean wing chord, and the experimental results are from Hinson and Burdges [102]. Note that the numerical results, including the shock strength and position, are in excellent agreement with experiment at all three span stations displayed.

Additional full potential flow solvers with three-dimensional capabilities include the codes of Chattot et al. [103], Eberle [104] and Chen and Caughey [105]. The first two codes are capable of simulating the flow about isolated wing geometries, and the latter code is capable of simulating the flow about axisymmetric inlets

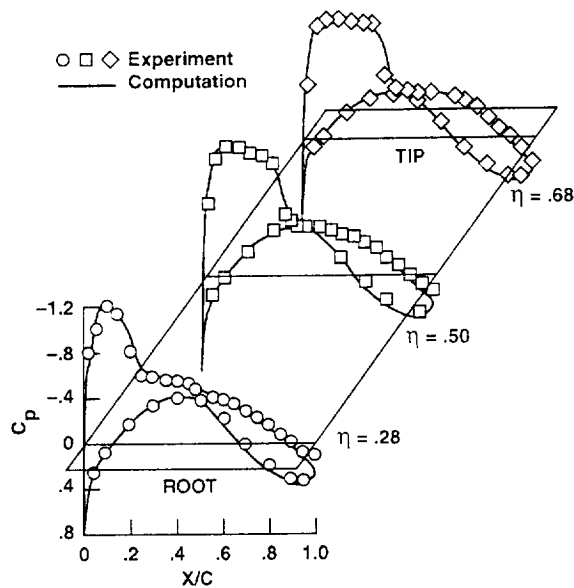


Fig. 14. Wing surface pressure coefficient distributions for a wing/fuselage configuration, $M_\infty = 0.819$, $\alpha = 1.96^\circ$, $Re_c = 6 \times 10^6$, taken from Street [97].

at angle of attack with and without center bodies. All three of these codes utilize the SLOR iteration scheme, although the first code also has an approximate factorization (AF) option (which is discussed in Section 3.4). All three of these codes solve the conservative form of the full potential equation, although the first and third codes also have independent nonconservative options.

3.3.2. Artificial density schemes

Another type of spatial discretization scheme for the full potential equation written in conservative form is the artificial density scheme. This discretization scheme has been independently developed in several different forms by Eberle [104], Holst and Ballhaus [106] and Hafez et al. [107]. These approaches, although not identical, have certain similarities which can be attributed to the earlier work of Jameson [92]. Jameson's work is characterized by a scheme with an explicitly added artificial viscosity term [see Eqs. (39) and (40)]. The artificial viscosity term is designed to provide an upwind bias for supersonic regions of flow, but does not affect the centrally differenced scheme in subsonic regions. The artificial density scheme uses this approach with one basic simplification: the upwind bias is accomplished by an upwind evaluation of the density. The three procedures compute this upwind density quantity in different ways.

In the procedure of Holst and Ballhaus [106] the finite-difference approximation for the full potential equation written in two-dimensional curvilinear coordinates [see Eq. (37)] is given by

$$\bar{\delta}_\xi \left(\frac{\bar{\rho} U}{J} \right)_{i+1/2,j} + \bar{\delta}_\eta \left(\frac{\bar{\rho} V}{J} \right)_{i,j+1/2} = 0, \quad (41)$$

where the density coefficients $\bar{\rho}$ and $\bar{\rho}$ are defined by

$$\bar{\rho}_{i+1/2,j} = [(1-v)\rho]_{i+1/2,j} + v_{i+1/2,j} \rho_{i+r+1/2,j}, \quad (42a)$$

$$\bar{\rho}_{i,j+1/2} = [(1-v)\rho]_{i,j+1/2} + v_{i,j+1/2} \rho_{i,s+1/2}. \quad (42b)$$

The r and s subscripts used above control the upwind direction of the density coefficients and are defined by

$$r = \begin{cases} +1, & U_{i+1/2,j} < 0, \\ -1 & U_{i+1/2,j} > 0, \end{cases} \quad s = \begin{cases} +1, & V_{i,j+1/2} < 0, \\ -1 & V_{i,j+1/2} > 0, \end{cases} \quad (42c)$$

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

$$v_{i-1/2,j} = \begin{cases} \max[(M_{i,j}^2 - 1)C, 0], & U_{i-1/2,j} > 0, \\ \max[(M_{i+1,j}^2 - 1)C, 0], & U_{i-1/2,j} < 0, \end{cases} \quad (42d)$$

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

The spatial differencing scheme described by Eqs. (41), (42a)-(42d) provides an upwind influence in supersonic

regions 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 with the Jameson–Caughey finite-volume scheme [see Eqs. (39) and (40)], the artificial density approach closely approximates the effects of a rotated differencing scheme. This aspect contributes to the stability and reliability of the overall algorithm and allows computations of many difficult strong-shock cases.

Another variation of the artificial density spatial differencing scheme has been presented by Hafez et al. [107]. In this scheme, which is sometimes called the artificial compressibility scheme, the density coefficients in both coordinate directions are defined by

$$\hat{\rho}_{i,j} = \rho_{i,j} - v_{i,j}(\rho_s \Delta s)_{i,j}, \quad (43a)$$

where

$$(\rho_s \Delta s)_{i,j} = \left(\frac{u}{q}\right)_{i,j} \bar{\delta}_x \rho \Delta x + \left(\frac{v}{q}\right)_{i,j} \bar{\delta}_y \rho \Delta y. \quad (43b)$$

The double-arrow notation indicates a first-order difference, always chosen to be in the upwind direction; s is the local streamwise coordinate direction; and v is a switching function defined similarly to Eq. (42d).

Many researchers have used one of the artificial density spatial discretization approaches mentioned above because of the simple, reliable way in which the supersonic region is stabilized. A few of these applications include Wong and Hafez [108] for airfoil computations; Farrell and Adamczyk [109] Akay and Ecer [110] and Deconinck and Hirsch [111] for cascade computations; Shankar [112] for supersonic space marching problems; Green and South [113] for axisymmetric computations; Eberle [114,115] for a variety of different applications; Steger and Caradonna [116] and Goorjian [117] for unsteady computations; Ecer and Spyropoulos [118] for wing-body computations; Neel [119] for wing computations; and Holst and Thomas [120] and Holst [121,122] for wing and wing/body computations.

3.3.3. Flux upwind schemes

Another type of supersonic flow stabilization method, similar to the artificial density approach, is the flux upwind scheme. This type of scheme has been reported by Engquist and Osher [123], Goorjian and Van Buskirk [124], Goorjian et al. [125], Boerstol [126], Slooff [127], Osher et al. [128,129] and Hafez et al. [130]. The basic idea is to evaluate the entire flux in the upwind direction using a special construction that will produce good shock capturing characteristics and smooth flow gradients through the sonic line, i.e., less shock smearing and no expansion shocks. There are several approaches

for this type of scheme. Following Van Leer [131] and Slooff [127] and using a one-dimensional discretized equation given by

$$(\rho \phi_x)_x \cong \frac{1}{\Delta x} [(\overline{\rho u})_{i+1/2} - (\overline{\rho u})_{i-1/2}],$$

the Godunov and Engquist–Osher flux upwind schemes are given by

$$(\overline{\rho u})_{i+1/2} = \rho^* u^* - \max(\Delta_{i-1/2}^+, \Delta_{i+1/2}^-) \quad (\text{Godunov}),$$

$$(\overline{\rho u})_{i+1/2} = \rho^* u^* - \Delta_{i-1/2}^+ - \Delta_{i+1/2}^- \quad (\text{Engquist-Osher}),$$

where

$$\Delta_{i-1/2}^+ = \begin{cases} \rho^* u^* - (\rho u)_{i-1/2} & \text{if } u_{i-1/2} > u^*, \\ 0 & \text{if } u_{i-1/2} < u^*, \end{cases}$$

$$\Delta_{i+1/2}^- = \begin{cases} 0 & \text{if } u_{i+1/2} > u^*, \\ \rho^* u^* - (\rho u)_{i+1/2} & \text{if } u_{i+1/2} < u^*. \end{cases}$$

In the above equations ρ^* and u^* are sonic values of the density and speed, respectively, and the overbar quantities represent the upwind evaluated fluxes. Both of these schemes produce standard discretizations in regions away from sonic lines and 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 in the above two schemes is in the shock point operator. An additional example of the flux upwind scheme derived from the approach described in Osher et al. [129] is presented in Section 3.10 in the context of a finite element method.

Results comparing a number of supersonic stabilization schemes including both artificial density and flux upwind schemes are presented and compared in Habashi and Hafez [132], Volpe and Jameson [133] and Dulikravich [134]. For several applications involving weak shock waves the flux upwind scheme is superior in shock capturing sharpness. For stronger transonic shocks the two approaches produce similar results. For convergence reliability and efficiency the two approaches (when using the same iteration scheme) are also similar.

3.3.4. Entropy and vorticity corrections

An interesting characteristic associated with the conservative full potential equation spatial-discretization scheme is the ability to add entropy corrections at shock waves. Because of the isentropic and irrotational approximations associated with all traditional potential formulations, shock wave capture is accurate only for weak shock waves (see discussion in Section 2.9). As the shock wave strength increases the error increases. Several different entropy correction procedures designed to approximately correct this problem are available including the techniques of Hafez and Lovell [135,136] and Klopfer

and Nixon [137]. The basic idea involved in this approach is to use a nonisentropic expression for the density given by (see Bridgeman et al. [138] for a derivation)

$$\rho = \rho_{\text{isen}} e^{-\Delta s/R},$$

where ρ_{isen} is the traditional isentropic density computed from (for example) Eq. (30b), R is the real gas constant, and Δs is the increase in entropy across the shock wave given by [32]

$$\frac{\Delta s}{R} = \frac{1}{\gamma - 1} \ln \left[\frac{2\gamma M_1^2 - (\gamma - 1)}{\gamma + 1} \right] - \frac{\gamma}{\gamma - 1} \ln \left[\frac{(\gamma + 1)M_1^2}{(\gamma - 1)M_1^2 + 2} \right].$$

In the above equation M_1 is the Mach number just upstream of the shock wave. In this implementation the flow is assumed to be steady, and the shock wave is assumed to be normal to the local velocity vector. Because the entropy convects with the flow, its material derivative is zero everywhere except at a shock wave. Thus, the full potential equation in its traditional isentropic form can be used everywhere except at shock waves (see Bridgeman et al. [138]). At the shock wave, the density is modified using the nonisentropic density formula. Identification of the shock location is simply achieved (for normal shocks) by finding each grid cell where the Mach number decreases through one in the positive flow direction. Hafez and Lovell [136] present an approach for applying an entropy correction to oblique shocks, which requires identification of the shock location and its angle with the local flow direction. However, difficulty in numerically identifying these quantities, coupled with the fact that entropy corrections for oblique shocks are not as important as for normal shocks, has generally dissuaded researchers from attempting entropy corrections for oblique shocks. Despite the steady flow assumption, this approach has been successfully applied to unsteady problems by assuming quasi-steady flow. It should be noted that the above approach assumes no correction for vorticity. A simple vorticity correction model consistent with the above entropy correction model is also possible. Two examples are presented in Kinney and Hafez [139] and Batina [140].

Numerous applications utilizing the entropy (and vorticity) correction procedure described above (or a variation thereof) have been presented in the literature for extending the range of applicability of various potential formulations. Examples of these applications include Siclari and Visich [141] and Siclari and Rubel [142] for supersonic marching solutions, Fuglsang and Williams [143] and Batina [140] for solutions of the unsteady TSD equation, Whitlow [144] for airfoil calculations, Bridgeman et al. [138,145] and Chen and Bridgeman [146] for unsteady helicopter rotor applications, Zi-qiang

and Xue-Song [46] for airfoil, wing and wing-body configurations, and Kinney and Hafez [139] for transonic wing computations. A numerical result showing differences between the traditional isentropic full potential approach and the same computation with an entropy correction is presented in Section 3.9. The above entropy (and vorticity) correction procedure allows the potential equation to compute flows with stronger shock waves and produce results that approximate the Euler equations.

3.3.5. Freestream consistency conditions

Grid generated irregularities, such as mapping singularities, rapid stretching, cell skewness, or grid coarseness, exist in many applications, and cause accuracy difficulties for most spatial discretization schemes, especially finite-difference schemes. Ideally, a stable flow-solver algorithm, which can handle all of the above-mentioned irregularities, yet provide uniform accuracy over the entire grid, is desired. In a spatial discretization formulation that utilizes the general mapping procedure described in Section 2.8, it can be shown that if the metric differencing is implemented properly, the truncation error associated with a freestream distribution of the dependent variable is zero. That is, freestream is admitted as a solution to the finite-difference equations. If the truncation error associated with freestream flow is zero, then accuracy for any type of solution will be improved. This type of procedure is addressed in Pulliam and Steger [147] for the Euler equations, but is not used because of the small improvements in accuracy obtained on smooth grids. Thomas and Lombard [148] and Hindman [149] also study geometrically induced errors associated with metric differencing and find that certain procedures are better than others.

All of the above work is associated with the Euler equations. Chattot et al. [103] present a spatial differencing scheme which provides perfect freestream capture for the full potential equation. However, the full potential equation is not written in conservative form, i.e., the metrics are outside the main flux differentiation. On smooth grids, where metric variation is small, this formulation behaves like conservative form. Flores et al. [150] present a freestream-preserving, spatial differencing scheme for the conservative full potential equation written in general curvilinear coordinates [see Eqs. (30a) and (30b) and the associated metric definitions]. Unlike the Euler equation scheme presented in Pulliam and Steger [147] which produces perfect freestream capture with a single consistency condition, the full potential equation requires three conditions. The first condition is associated with the density calculation and is developed as follows (assuming two-dimensional steady flow). The density can be written solely as a function of fluid speed. Thus, the numerical prediction of freestream density requires the prediction of the freestream fluid speed. The

fluid speed can be written as (using the physical domain metrics from Section 2.8)

$$q^2 = J^2[(y_\eta \phi_\xi - y_\xi \phi_\eta)^2 + (-x_\eta \phi_\xi + x_\xi \phi_\eta)^2].$$

Analytically speaking this expression reduces to q_∞^2 for a freestream distribution of ϕ in a trivial fashion. The key question is: what is the value of q for a freestream distribution of ϕ when difference formulas are used to replace all derivatives? The numerical evaluation of this expression reduces 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 the above equation and then using the exact freestream ϕ distribution to simplify. This is the first freestream consistency condition.

The second and third consistency conditions are associated with the flux calculation. For freestream flow the numerically computed density is a constant (assuming the density consistency condition is satisfied). Thus, the full potential equation can be written as

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

where ϕ_x and ϕ_y are given by

$$\phi_x = J(y_\eta \phi_\xi - y_\xi \phi_\eta) = u, \quad \phi_y = J(-x_\eta \phi_\xi + x_\xi \phi_\eta) = v.$$

If the difference operators for the ξ -differences involving x , y and ϕ in each flux computation are the same, and if the difference operators for all η -differences involving x , y and ϕ in each flux computation are the same, respectively, then the numerically evaluated values of ϕ_x and ϕ_y (with a freestream distribution of ϕ) are equal to u_∞ and v_∞ . This is the second freestream consistency condition. Note that the first and second freestream consistency conditions are the same providing the density and flux computations are performed in the same locations. In general they are not, and thus, these two freestream consistency conditions must be considered as separate conditions. As a consequence, the density and flux metrics must be computed and stored separately.

With $\phi_x = u_\infty$ and $\phi_y = v_\infty$ the full potential equation can be further simplified to

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

For this equation to be satisfied numerically, the finite-difference operators used for the flux metrics must commute with the finite-difference operators used for the flux derivatives. This is the third freestream consistency condition. This last condition is the same condition presented in Pulliam and Steger [147] and is required (by itself) to achieve perfect freestream capture for the Euler equations. Extension of these metric-differencing require-

ments to three dimensions for the full potential equation is straightforward but tedious (see Flores et al. [150] or Thomas and Holst [151]). An additional paper addressing accurate three-dimensional metric evaluation for the full potential equation using a finite-difference approach is Jiang and Cai [152].

3.4. Approximate factorization iteration schemes

The vast majority of all full potential solvers so far presented have utilized the SLOR iteration scheme. Other iteration schemes, including approximate factorization (AF) and multi-grid schemes, have superior convergence characteristics, i.e., solutions are obtained with fewer iterations and less computer time. Approximate factorization iteration schemes can be examined by considering the following general two-level iteration procedure

$$NC^n + \omega L\phi^n = 0, \quad (44)$$

where $C^n (= \phi^{n+1} - \phi^n)$ is the correction, $L\phi^n$ is the residual, which is a measure of how well the discretized approximation to the governing PDE is satisfied by the n th iterate of the dependent variable ϕ , and ω is a relaxation parameter. The iteration scheme given by Eq. (44) can be considered to be an iteration in pseudotime, where the n superscript indicates the time-step level of the solution, i.e., $(\cdot)^{n+1} - (\cdot)^n \sim \Delta t(\cdot)$. The operator N determines the type of iterative procedure, and therefore, determines the rate at which the solution procedure converges.

Classical successive over-relaxation (SOR) or SLOR schemes effectively use only a portion of the L operator in forming the N operator. As a consequence, the iteration scheme is relatively simple, but the convergence rate is relatively slow. In the AF approach, the philosophy is to choose a representation for N that closely approximates L . This, in theory, will produce a scheme with good convergence characteristics. The procedure for obtaining N consists of two steps: (1) linearize L and (2) factor the linearized result. There are usually two factors for two-dimensional algorithms and three factors for three-dimensional algorithms. The resulting scheme retains the simplicity of requiring only narrow-banded scalar matrix operations. The effects of both the factorization error terms and the linearization are removed from the solution simultaneously by means of the iteration scheme. Because each grid point is influenced by every other grid point during each iteration, much faster convergence is obtained.

Several early examples of AF schemes can be found in Peaceman and Rachford [153], Douglas [154], Douglas and Gunn [155] and Yanenko [156]. In these pioneering applications different forms of the AF scheme are introduced and applied to purely parabolic or elliptic

equations. Additional classical information regarding AF schemes can be found in Mitchell [27]. The first application of the AF approach to transonic flows governed by potential formulations is the work of Ballhaus and Steger [1]. Since this work many calculations using AF iteration schemes have been obtained including Goorjian [117], Steger and Caradonna [116] and Sankar and Tassa [157] for time-accurate full potential applications; Ballhaus et al. [158] for steady two-dimensional TSD computations; Holst and Ballhaus [106], Holst [159], Chattot and Coulombeix [160], Deconinck and Hirsch [111], Roach and Sankar [161] and Baker [162] for steady two-dimensional full potential computations; and Chattot [163], Baker and Forsey [164], Sankar et al. [165] and Holst and Thomas [120] for three-dimensional full potential computations.

Two widely used AF schemes are now presented in more detail. In the interest of brevity only two-dimensional versions will be considered, both designed to solve the conservative full potential equation given by Eqs. (37). The first scheme is a reformulation of the Peaceman–Rachford alternating direction implicit (ADI) scheme and can be expressed by choosing N [from Eq. (44)] as follows:

$$NC_{i,j}^n = -\frac{1}{\alpha}(\alpha - \bar{\delta}_\xi A_i \bar{\delta}_\xi)(\alpha - \bar{\delta}_\eta A_j \bar{\delta}_\eta)C_{i,j}^n, \quad (45)$$

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

$$A_i = \left(\frac{\bar{\rho}A_1}{J}\right)_{i-1/2,j}, \quad A_j = \left(\frac{\bar{\rho}A_2}{J}\right)_{i,j-1/2}. \quad (46)$$

In the above expressions, the density coefficients, $\bar{\rho}$ and $\bar{\rho}$, are defined by Eqs. (42a) and (42b) and A_1 , A_2 and J are metric quantities defined similarly to the metrics in Eqs. (31b). The ADI scheme represented by Eq. (45) is implemented in a two-sweep format given by

Sweep 1:

$$(\alpha - \bar{\delta}_\xi A_i \bar{\delta}_\xi)f_{i,j}^n = \alpha \omega L\phi_{i,j}^n, \quad (47a)$$

Sweep 2:

$$(\alpha - \bar{\delta}_\eta A_j \bar{\delta}_\eta)C_{i,j}^n = f_{i,j}^n. \quad (47b)$$

In Eqs. (47a) and (47b), $f_{i,j}^n$ is an intermediate result stored over the entire finite-difference grid and the residual $L\phi_{i,j}^n$ is defined by Eq. (41). Sweep 1 consists of a set of tridiagonal matrix equations along the ξ -coordinate direction, and sweep 2 consists of a set of tridiagonal matrix equations along the η -coordinate direction. The construction of this ADI scheme does not automatically provide the necessary ϕ_{st} temporal damping required to stabilize supersonic flow regions. However, this type of term can be included by adding

$$\mp \beta_\xi |U_{i,j}| \bar{\delta}_\xi \quad \text{and} \quad \mp \beta_\eta |V_{i,j}| \bar{\delta}_\eta$$

inside the parentheses of the first and second sweeps, respectively. The double-arrow notation on these operators 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 contravariant velocity component scaling used in the above expressions provides a smooth transition from forward to backward differencing when the flow direction changes sign. The β_ξ and β_η coefficients are constants specified by the user.

The ADI scheme presented above is stable providing $0 \leq \omega \leq 2$ and $\alpha \geq 0$. Because the only condition for stability on the α parameter, which behaves like the inverse of the physical time step, is that it be positive, the ADI scheme is said to have unconditional linear stability, as expected for a fully implicit scheme. The best strategy for obtaining rapid convergence is to use a repeating sequence of α values. The small values reduce the low-frequency errors and the large values act as a smoothing mechanism, and thus, reduce the high-frequency errors. A suitable sequence is given by

$$\alpha_k = \alpha_H \left(\frac{\alpha_L}{\alpha_H}\right)^{(k-1)/(M-1)}, \quad k = 1, 2, \dots, M, \quad (48)$$

where M is the number of elements in the sequence and α_L and α_H are the sequence endpoints corresponding to the low- and high-frequency limits, respectively. In practice, α_L and α_H are often “optimized” by trial-and-error numerical experimentation. This typically is performed only once for each code and grid size, as optimal values of α_L and α_H do not strongly depend on solution characteristics. For more information on the optimal choice of these parameters, the interested reader is referred to Catherall [166] where a detailed analysis is performed for several AF iteration schemes.

Faster convergence can be obtained with the ADI iteration algorithm. An estimate for the number of iterations N_{its} required to drop the error by N_e orders of magnitude for solving Laplace’s equation (with optimal acceleration parameters) is given by [26]

$$N_{its} \sim -0.645 N_e \log\left(\frac{\Delta}{2}\right), \quad (49)$$

where Δ is the grid spacing used in the computation. Comparing this estimate with those established in Eqs. (35) and (36) for the line Gauss–Seidel and SLOR iteration schemes, respectively, it is easy to see that the ADI scheme is superior in convergence efficiency, being as much as an order of magnitude faster for fine grid computations.

The second AF scheme described is the so-called AF2 scheme, which was first studied in Ballhaus and Steger [1]. This iteration algorithm was subsequently used to solve the steady TSD equation by Ballhaus et al. [158] and the conservative full potential equation by Holst and

Ballhaus [106]. The AF2 fully implicit scheme can be expressed by choosing the N -operator of Eq. (44) as follows:

$$NC_{i,j}^n = -\frac{1}{x}(x - \bar{\delta}_\xi A_i)(x\bar{\delta}_\xi - \bar{\delta}_\eta A_j \bar{\delta}_\eta)C_{i,j}^n, \quad (50)$$

where (as with the ADI scheme) A_i and A_j are defined by Eqs. (46) and x is an acceleration parameter defined by Eq. (48). This scheme is implemented in a two-sweep format given by

Sweep 1

$$(x - \bar{\delta}_\xi A_i)f_{i,j}^n = x\omega L\phi_{i,j}^n, \quad (51a)$$

Sweep 2

$$(x\bar{\delta}_\xi - \bar{\delta}_\eta A_j \bar{\delta}_\eta)C_{i,j}^n = f_{i,j}^n. \quad (51b)$$

In Eqs. (51a) and (51b), $f_{i,j}^n$ is an intermediate result stored over the entire finite-difference grid and the residual $L\phi_{i,j}^n$ is defined by Eq. (41). Sweep 1 consists of a set of bidiagonal matrix equations along the ξ -coordinate direction, and sweep 2 consists of a set of tridiagonal matrix equations along the η -coordinate direction. With the AF2 factorization, the ξ -difference approximation is split between the two sweeps. This generates a ϕ_{ξ} -type term, which is useful to the iteration scheme in supersonic flow regions as time-like dissipation. The split ξ term also places a sweep direction restriction on both sweeps, namely, in the negative ξ direction for the first sweep [Eq. (51a)], and in the positive ξ direction for the second sweep [Eq. (51b)]. Flow direction imposes no sweep direction limitations on either of the two sweeps.

For curvilinear grid applications when the ξ direction is not aligned with the local flow direction, additional temporal damping terms aligned with the η -coordinate direction can be included by adding $\mp x\beta\bar{\delta}_\eta$ inside the parentheses of sweep 2. Again, the double arrow notation is used to indicate that the difference direction must always be upwind, and the sign is chosen so as to increase the magnitude of the matrix diagonal coefficient. The parameter β is a user-specified constant, which only needs to be activated in supersonic regions of flow when the flow direction is (to a significant extent) along the η direction.

Another form of the AF2 scheme that splits the η difference operator between the two factors is possible (see for example the AF2 variation in Holst [159]). This type of AF2 scheme is attractive for some applications involving general curvilinear grids, e.g., airfoil or wing computations using "O" grid topologies. More recent implementations of the AF2 scheme for transonic potential flow computations include South and Hafez [167] for airfoil computations; Vadyak and Atta [168] for three-dimensional nacelle analysis; Jialin [169] for three-dimensional axial-flow compressor flows; Holst [122] for chimera zonal grid applications; and Cosentino and

Holst [170], Cheung and Holst [171], de Mattos [172] and de Mattos and Wagner [173] for analysis and design of transonic wings. More on the use of transonic potential formulations in numerical optimization and design applications will be presented in Section 3.10.

3.5. Convergence characteristics of SLOR, ADI and AF2

Numerical results comparing the convergence characteristics of the two fully implicit algorithms described above (ADI and AF2) with the classical SLOR iteration algorithm are now presented. All three iteration schemes are 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 ($M_\infty = 0.7$ and 0.84). Pressure coefficient distributions for these two cases are displayed in Fig. 15. 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 Holst and Ballhaus [106].

Convergence characteristics for the subcritical case are displayed in Fig. 16. 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 and in terms of iteration count, the ADI scheme is about twice as fast as AF2 and about 16 times faster than SLOR. However, the ADI and AF2 schemes take about 50 and 30% more CPU time per iteration than SLOR, respectively, which should be considered when speed ratios based on the total amount of computational work are desired. Convergence

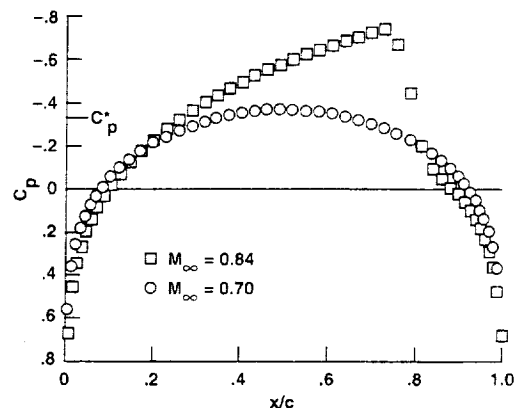


Fig. 15. Pressure coefficient distributions from subcritical ($M_\infty = 0.7$) and supercritical ($M_\infty = 0.84$) flow computations about a 10%-thick circular-arc airfoil, taken from Holst and Ballhaus [106].

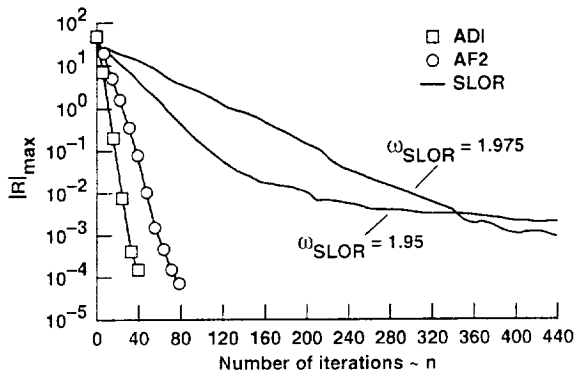


Fig. 16. Maximum residual convergence history comparison for the subcritical case, $M_\infty = 0.7$, taken from Holst and Ballhaus [106].

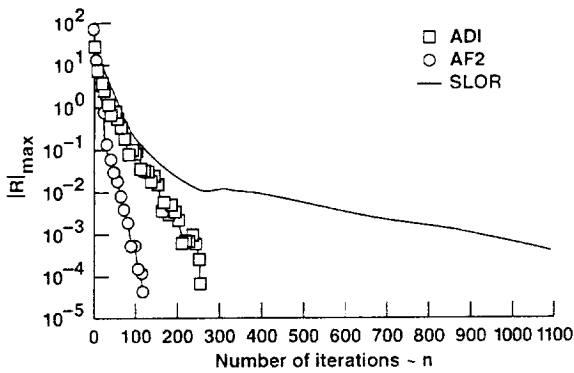


Fig. 17. Maximum residual convergence history comparison for the supercritical case, $M_\infty = 0.84$, taken from Holst and Ballhaus [106].

characteristics for the supercritical case are displayed in Fig. 17. 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 Fig. 18. The AF2, ADI and SLOR schemes reach the final value of NSP in 29, 103 and 320 iterations, respectively.

The AF2 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 [81].

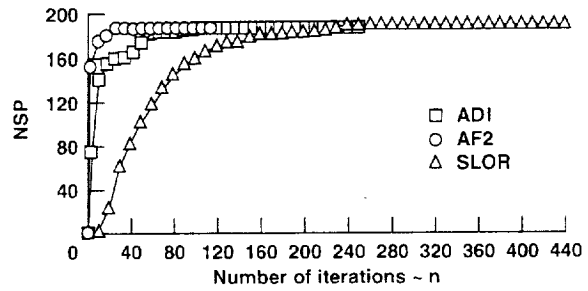


Fig. 18. Development of the supersonic region, $M_\infty = 0.84$, taken from Holst and Ballhaus [106].

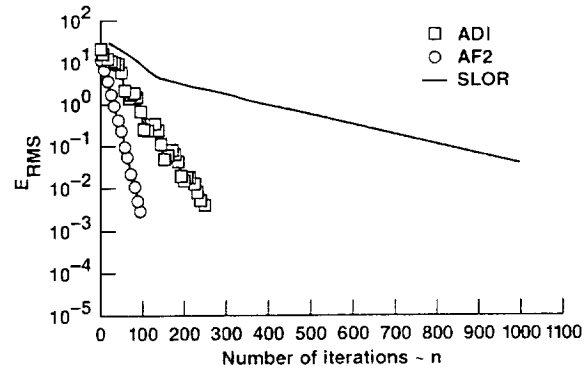


Fig. 19. Maximum RMS error convergence history comparison for the supercritical case, $M_\infty = 0.84$, taken from Holst and Ballhaus [106].

The convergence histories displayed above involve plotting residual versus iteration. Another more appropriate means of studying an iteration scheme's convergence properties is to look at error versus iteration. Such a plot is displayed in Fig. 19. In this case the error plotted on the vertical axis is the RMS error in the surface pressure coefficient, which is computed from

$$E_{RMS}^n = \left[\frac{1}{NI} \sum_{i=1}^{NI} (c_p^n - \bar{c}_p)^2 \right]^{1/2},$$

where c_p^n is the surface pressure coefficient at the i th grid point and the n th iteration, \bar{c}_p is the surface-pressure coefficient at the i th grid point taken from the tightly converged solution, and NI is the total number of surface grid points. By comparing the residual history curves (Fig. 17) with the error history curves (Fig. 19), it can be seen that reducing the maximum residual by a fixed amount (for example by two orders of magnitude) for the AF2 and SLOR schemes, does not result in equal levels of error reduction. The SLOR residual drops very rapidly initially and then levels off. The SLOR RMS error drops continuously but very 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 only one order of magnitude. In contrast, both the maximum residual and RMS error results for the ADI and AF2 schemes are nearly straight lines with about the same slopes.

This behavior is the result of two factors (Holst and Ballhaus [106]). First, the (optimized) ADI and AF2 schemes treat all error components equally well (approximately), whereas the SLOR scheme (even in optimal form) performs efficiently on only the high-frequency error components, i.e., it is a good smoothing algorithm. Second, it can be shown that the residual is a weighted sum of all the errors present in a solution, but with weighting factors heavily biased toward the highest-frequency errors. Thus, in the early stages of convergence as SLOR is smoothing the high-frequency errors typically caused by an impulsively started freestream initial condition, the residual drops rapidly, while the error, paced by low-frequency components, drops slowly. As a result of this behavior, the maximum residual should never be used to compare the convergence properties of two dissimilar iteration schemes. The RMS error is much better suited for this purpose. In practice, using the maximum residual to monitor convergence 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.

3.6. Multigrid iteration schemes

The multigrid scheme was originally developed for solving elliptic equations, but has subsequently been applied to a much wider range of problems, including hyperbolic problems with shocks, time-accurate problems, and problems of mixed type such as those associated with transonic flow. This scheme is actually a convergence acceleration technique and requires a base iteration scheme, e.g., SOR, SLOR, or AF. Multigrid-like schemes have existed for quite some time, having been first introduced by Fedorenko [174]. Since then, several authors have analyzed the technique, including Bakhalov [175], Nicolaidis [176] and Hackbusch [177]. The most significant aspect of a multigrid scheme is fast convergence. Fast convergence is produced by using a sequence of grids ranging from very coarse to 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 grid. Many relaxation schemes exist that work well on high-frequency errors, e.g., most AF schemes with properly chosen acceleration parameters. A suitable relaxation scheme 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 grid is actually the lowest-frequency error existing in the problem. Because this usually troublesome

low-frequency error is efficiently dealt with on a coarse grid, 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 now described in general terms. Suppose a solution is desired to the following equation:

$$L^h \phi = f,$$

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

$$\phi = u + v,$$

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

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

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

$$u^{\text{new}} = u + I_{2h}^h v,$$

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

Brandt [178,179] presents early numerical applications of the multigrid scheme. In the latter reference a good historical review of early multigrid schemes is presented. The first use of the multigrid scheme for transonic calculations is presented by South and Brandt [180]. In this study, numerical solutions of the TSD potential equation for non-lifting airfoils are obtained. The multigrid-enhanced-SLOR scheme is found to be a factor of three faster than an optimized SLOR scheme on uniform grids and a factor of two faster on stretched grids. A primary difficulty involved the existence 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 their multigrid implementation did not have uniform smoothing properties in both directions, especially for highly-stretched grids and suggested the use of an ADI-type scheme as the base algorithm.

Another approach proposed by Arlinger [181] is to refine or coarsen the grid in only one coordinate direction while doing line relaxation along the opposite direction. This technique produces convergence rate acceleration but does not take full advantage of the multigrid philosophy. Another successful implementation of multigrid to transonic flow is the work of Jameson [182]. In this work, the two-dimensional full potential equation in conservative form is solved using multigrid with an AF base iteration scheme. The Jameson multigrid algorithm uses a recursive approach, instead of the adaptive approach advocated by Brandt [178,179]. In the adaptive multigrid approach, the decision to proceed to the next grid, either coarser or finer, is based on a convergence rate criterion. If the solution residual is dropping slowly, the iteration proceeds to coarser grids. If the solution residual is dropping rapidly, the iteration proceeds to finer grids. In the recursive approach, a single multigrid cycle starts with a fine grid iteration, followed by an iteration on the second finest grid, etc. This continues until the coarsest grid is reached. Then the process is reversed, starting with the coarsest grid and ending with the second finest grid. Therefore, one multigrid cycle consists of one iteration on the finest grid and two iterations on every other grid. If a fine grid iteration is defined as a unit of work, then one multigrid cycle, using the recursive approach, requires about $1\frac{2}{3}$ work units (for two-dimensional problems) plus interpolation operations.

Computed transonic airfoil results produced by the Jameson multigrid scheme are displayed in Figs. 20 and 21. The pressure coefficient distribution for this case (an NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$), is displayed in Fig. 20. A moderate-strength shock exists at about 60% of chord. Convergence histories for this case, computed using different numbers of grids (from one grid, i.e., no multigrid, up to five grids) are shown in Fig. 21. Notice that the maximum residual has been reduced below 10^{-8} (for the five-grid case), which corresponds to an eight order-of-magnitude reduction. This is achieved in just 29 multigrid cycles (approximately 50 work units). The convergence rate parameter (CR), which is defined as the mean reduction in the average residual per unit of work, is also displayed in Fig. 21 for each convergence history curve. Increasing the number of grids greatly improves the convergence rate.

Other researchers have used the multigrid algorithm to solve the full potential equation in a variety of applications including Fuchs [183], Boerstoel [126], Deconinck and Hirsch [184] and Sankar [185] for two-dimensional calculations; Arlinger [186] for axisymmetric calculations; Volpe [187] for two-element airfoil computations; McCarthy and Reyhner [188] and Brown [189] for three-dimensional engine-inlet calculations; Shmilovich and Caughey [190] and Caughey [191] for three-dimensional wing calculations; Chen et al. [192]

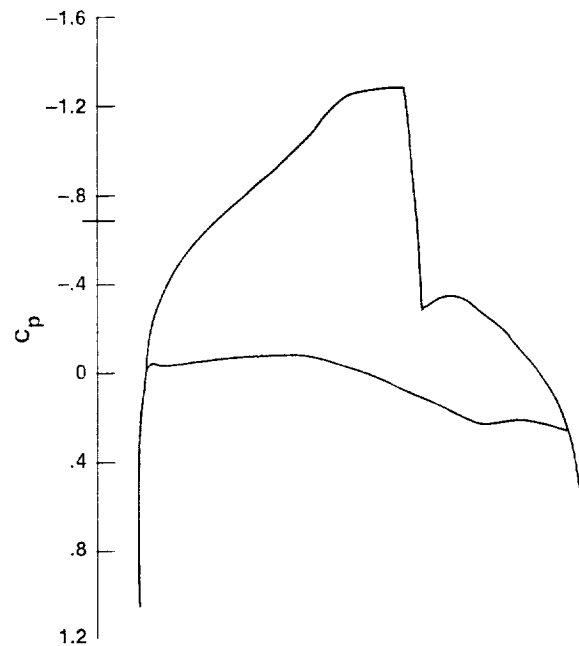


Fig. 20. Converged airfoil pressure coefficient distribution obtained from multigrid code, taken from Jameson [182].

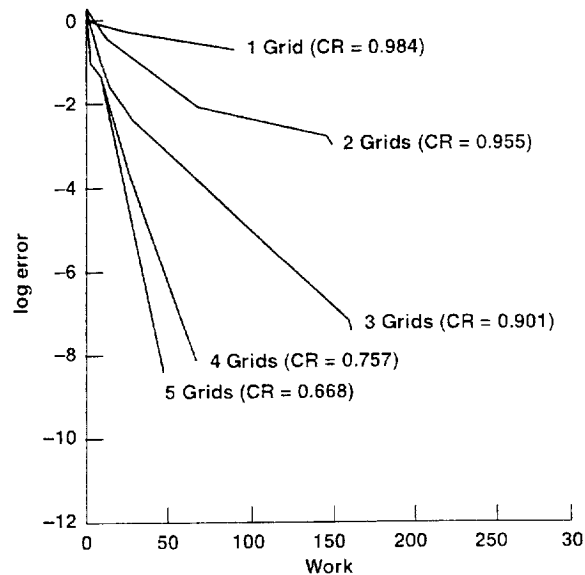


Fig. 21. Maximum residual convergence histories showing effect of the number of grid levels on multigrid convergence, taken from Jameson [182].

for wing-body calculations; Chen et al. [101] for fuselage-pylon-nacelle calculations with and without power; van der Vooren et al. [193] and van der Vooren and van der Wees [194] for wing and wing-body computations with special emphasis on accurate drag prediction; and

Shmilovich and Caughey [195] for wing/body/tail computations. The multigrid-based procedure of Shmilovich [196] designed to solve general geometry inlets at incidence and yaw is quite interesting. It consists of a base SLOR scheme that alternates between the streamwise and normal-like coordinate directions and contains a direct/inverse boundary layer correction procedure due to Cebeci et al. [197]. Several transonic flows with boundary layer separation are presented that demonstrate the method's capabilities.

3.7. Other iteration schemes

Sankar and Tassa [157] have applied the strongly implicit procedure (SIP) introduced by Stone [2] to the numerical solution of the full potential equation for unsteady transonic airfoil calculations. Additional steady applications include those of Sankar et al. [198] for transonic wing computations, Roach and Sankar [161] for transonic cascade computations, and Sankar [185] and Gordon and Arieli [199] for airfoil computations. A multigrid-enhanced SIP approach is used in van der Vooren et al. [193] and van der Vooren and van der Wees [194] for wing and wing/body computations. 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 Malone and Sankar [200] for unsteady airfoil computations and Sankar et al. [165] and Sankar and Malone [201] for unsteady wing calculations.

South et al. [202] describe an algorithm called Zebra II, which is highly parallelizable and requires about the same number of iterations to converge as SLOR. This algorithm is an explicit or point scheme that mimics full-plane SOR. Iteration schemes that have superior convergence properties, and thus produce solutions in a minimal number of iterations, are available. A few examples include the conjugate-gradient methods of Bristeau et al. [203], Glowinski et al. [204], Chattot and Coulombeix [160] and Wong and Hafez [205] and the minimum residual method of Wong and Hafez [206,108] and Wong [207]. The work of Wong and Hafez [206] 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, and conjugate-gradient schemes with several types of preconditioning combined with both SLOR and Zebra. These iteration schemes are combined with two different 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 in computational efficiency by as much as a factor of ten for subcritical cases and by at least a factor of two for tough transonic cases.

3.8. Space marching schemes

Up to this point only schemes that are designed to solve transonic flows with subsonic freestream Mach numbers have been considered. Another class of problems equally amenable to full potential algorithms are those with low supersonic freestream Mach numbers. A central feature of this class of schemes is that the solution is marched, i.e., without global iteration, from upstream to downstream. Each cross-sectional plane solution is transonic-flow-like in that both subsonic and supersonic cross-flow Mach numbers may exist. Each cross-flow plane is iterated until convergence. Then the solution technique moves on to the next downstream plane. Because there is no global iteration, these so-called "space-marching" schemes are extremely efficient. For the full potential formulation to be valid for supersonic freestream flows, the resulting bow shock must be attached, i.e., blunt body problems are generally not included, and the normal component of the bow-shock Mach number must be at or below a value of about 1.3 (see Fig. 6). Thus, sharp, thin bodies at small angles of attack are generally the target application. If the body is thin enough, e.g., a five-degree cone, at a small enough angle of attack, e.g., 10° or less, freestream Mach numbers at or above two can easily be accommodated. Another characteristic of this type of flow is the existence of local pockets of subsonic flow. Such a problem requires a hybrid marching-relaxation iteration scheme.

Early full potential equation marching algorithms are described in Grossman [208], Grossman and Siclari [209], Siclari [210], and Siclari and Visich [141]. In these studies the nonconservative full potential equation is solved using an SLOR-like algorithm in each cross-flow plane. Grossman [208] and Siclari and Visich [141] use a conical-flow assumption to reduce the problem to two dimensions, i.e., transonic flow in the cross-flow plane. In conical flow all flow variables are assumed constant along "conical lines", i.e., straight lines that pass through the apex of the geometry. Thus, flow variable derivatives with respect to the conical direction are zero. This common assumption has been used by others (Bradley et al. [211] Sritharan and Seebass [212]) for solving the conservative form of the full potential equation. This approach, while retaining many features of three-dimensional supersonic flow, has the limitation that only conical-type bodies can be analyzed. The other methods presented above [209,210] are more general, utilizing a conformal-type mapping for grid generation in each cross-sectional plane, and thus, are generally applicable for nonconical geometries. The work in Siclari [210] and Siclari and Visich [141] is interesting in that shock fitting is utilized to accurately compute shock waves, both the bow shock and the cross-flow shock. In Siclari and Visich [141] the shock-jump conditions used in the shock-fitting scheme are nonisentropic (see Section 3.3), and thus

closely model the Euler equations. Results presented in the above references include cones, elliptic cones, and a simplified wing-body configuration. Good agreement with experimental surface pressures is obtained, including the capture of transonic-like cross-flow shock waves, providing viscous effects are not important. Additional applications involving more complete geometries include Walkley and Smith [213] for fighter forebodies with and without canopies and Rose et al. [214] for wing-body configurations. Computational times for this type of technique range from a few seconds to a minute on moderate-speed desk-top-type computers.

A space marching approach for solving the conservative full potential equation is presented in Shankar [112] and Shankar and Osher [215]. In this approach a locally iterated approximate factorization scheme is used to obtain each cross-flow-plane solution. This scheme (presented now in more detail) is designed to solve the steady full potential equation written in general coordinates given by

$$\left(\frac{\rho U}{J}\right)_{\xi} + \left(\frac{\rho V}{J}\right)_{\eta} + \left(\frac{\rho W}{J}\right)_{\zeta} = 0, \quad (52a)$$

$$\rho = \left[1 - \frac{\gamma - 1}{2} M_{\infty}^2 (U\phi_{\xi} + V\phi_{\eta} + W\phi_{\zeta} - 1)\right]^{1/(\gamma - 1)}. \quad (52b)$$

The above full potential governing equation is the same as Eq. (30a) except all the time terms have been eliminated. The above density relation is the same as Eq. (30b) except the time terms have been eliminated and the nondimensionalization is in terms of ρ_{∞} and q_{∞} instead of ρ_{∞} and a_{∞} . The metric and contravariant velocity component definitions given in Eqs. (31a) and (31b) are still valid for Eqs. (52a) and (52b).

A suitable space marching scheme is developed as follows. First, it is assumed that the ξ -coordinate direction is approximately aligned with the stream direction, and thus, this direction is the marching direction. Given a solution at $i, i - 1$, etc., the marching scheme is devised to compute the solution at the $(i + 1)$ st cross-sectional surface. The first step in building such a marching scheme is to linearize the first term in Eq. (52a). This is accomplished by noting that $\rho U/J = f(\phi)$ and by using a Taylor series as follows:

$$\left(\frac{\rho U}{J}\right)_{i-1} = \left(\frac{\rho U}{J}\right)_i + \frac{1}{J}(\rho_{\phi} U + \rho U_{\phi})_i \Delta\phi_i + O(\Delta^2), \quad (53)$$

where $\Delta\phi_i = \phi_{i-1} - \phi_i$ and the quantities ρ_{ϕ} and U_{ϕ} are operators given by

$$\rho_{\phi} = -\frac{\rho}{a^2} \left(U \frac{\partial}{\partial \xi} + V \frac{\partial}{\partial \eta} + W \frac{\partial}{\partial \zeta} \right),$$

$$U_{\phi} = A_1 \frac{\partial}{\partial \xi} + A_4 \frac{\partial}{\partial \eta} + A_5 \frac{\partial}{\partial \zeta}.$$

Substitution of the above two expressions into Eq. (53) yields after simplification the final linearization for the ξ -direction flux given by

$$\begin{aligned} \left(\frac{\rho U}{J}\right)_{i-1} &\cong \left(\frac{\rho U}{J}\right)_i + \left(\frac{\rho}{J}\right)_i \left[\left(A_1 - \frac{U^2}{a^2} \right) \frac{\partial}{\partial \xi} \right. \\ &\quad \left. + \left(A_4 - \frac{UV}{a^2} \right) \frac{\partial}{\partial \eta} + \left(A_5 - \frac{UW}{a^2} \right) \frac{\partial}{\partial \zeta} \right] \Delta\phi_i, \end{aligned} \quad (54)$$

where the only quantity in Eq. (54) that is evaluated at $i + 1$ is in $\Delta\phi_i$. All other quantities are evaluated at i . The first term in Eq. (52a) is discretized using a first-order upwind formula given by

$$\left(\frac{\rho U}{J}\right)_{\xi} \cong \frac{1}{\Delta \xi} \left[\left(\frac{\rho U}{J}\right)_{i-1} - \left(\frac{\rho U}{J}\right)_i \right].$$

A second-order discretization is also presented in Shankar and Osher [215], but is not discussed here. The nonlinearization given by Eq. (54) must be used for the ξ -direction fluxes at both $i + 1$ and i in order to maintain conservative form.

To complete the formulation of the marching scheme, linearizations for the other two terms in Eq. (52a) are required. These linearizations are accomplished by using a variation of the artificial density scheme discussed in Section 3.3. For example, the second term in Eq. (52a) is evaluated using

$$\begin{aligned} \left(\frac{\rho V}{J}\right)_{\eta} &\cong \frac{\partial}{\partial \eta} \left[\bar{\rho} \left(\frac{A_4}{\Delta \xi} + A_2 \frac{\partial}{\partial \eta} + A_6 \frac{\partial}{\partial \zeta} \right) \Delta\phi_i \right. \\ &\quad \left. + \frac{\bar{\rho}}{J} \left(A_2 \frac{\partial}{\partial \eta} + A_6 \frac{\partial}{\partial \zeta} \right) \phi_i \right]. \end{aligned} \quad (55)$$

The density coefficient $\bar{\rho}$ used in the above equation is similar to the density coefficient defined in Eqs. (42a) and (42b). In this algorithm it is given by

$$\begin{aligned} \bar{\rho}_{i+1,j+1/2,k} &= (1 - v_{i,j+1/2,k}) \rho_{j+1/2,k} \\ &\quad + \frac{1}{2} v_{i,j+1/2,k} (\rho_{j+2m,k} + \rho_{j-1+2m,k}), \end{aligned} \quad (56)$$

where m and v are defined by

$$m = \begin{cases} 0 & \text{when } V_{i,j+1/2,k} > 0, \\ 1 & \text{when } V_{i,j+1/2,k} < 0, \end{cases}$$

$$v_{i,j+1/2,k} = \mu \left(1 - \frac{A_2 a^2}{V^2} \right)_{i,j+1/2,k}.$$

The quantity μ appearing in the above equation is defined by

$$\mu = \begin{cases} 0 & \text{for } (A_2 - V^2/a^2) > 0 \text{ elliptic crossflow,} \\ 1 & \text{for } (A_2 - V^2/a^2) < 0 \text{ hyperbolic crossflow.} \end{cases}$$

With this definition of μ the density coefficient upwinding is smoothly switched on or off based on whether the

cross-flow direction is elliptic or hyperbolic. All the ϕ -derivatives appearing in Eq. (55) are centrally differenced. The density values used in Eq. (56) are computed in two ways depending on the problem being solved. For conical flow problems, all density values are evaluated using the previous marching plane i . For nonconical flow problems, the density values are initialized to the density values in the previous marching plane, and then updated via local iteration as the solution at $i + 1$ converges. Treatment of the third term appearing in Eq. (52a) is similar to the second term just discussed.

Next, the linearizations need to be combined to create the final iteration scheme used to advance the solution from marching plane i to marching plane $i + 1$. After combination, simplification and factorization the following iteration scheme is obtained

$$\left[1 + \frac{a_2}{\beta \Delta \xi} \frac{\partial}{\partial \eta} + \frac{1}{\beta} \frac{\partial}{\partial \xi} \frac{\partial A_4}{\partial \xi} + \frac{1}{\beta} \frac{\partial}{\partial \eta} \frac{\partial A_2}{\partial \xi} \right] \times \left[1 + \frac{a_3}{\beta \Delta \xi} \frac{\partial}{\partial \xi} + \frac{1}{\beta} \frac{\partial}{\partial \xi} \frac{\partial A_5}{\partial \xi} + \frac{1}{\beta} \frac{\partial}{\partial \xi} \frac{\partial A_3}{\partial \xi} \right] \Delta \phi_i = R_{i,j,k}, \tag{57}$$

where

$$\beta = \frac{\rho}{(\Delta \xi)^2 J} \left(A_1 - \frac{U^2}{a^2} \right), \quad a_2 = \frac{\rho}{J} \left(A_4 - \frac{UV}{a^2} \right),$$

$$a_3 = \frac{\rho}{J} \left(A_3 - \frac{UV}{a^2} \right).$$

Eq. (57) has the form

$$L_\eta L_\xi \Delta \phi = R,$$

where each of the factors on the left-hand side represent a set of scalar tridiagonal matrix equations. The inversion of these matrix equations is achieved in a two-step sequence given by

$$L_\eta (\Delta \phi)^* = R, \quad L_\xi (\Delta \phi) = (\Delta \phi)^*.$$

For the above marching scheme to be stable, it can be shown that the following condition is required

$$A_1 - \frac{U^2}{a^2} < 0.$$

This condition states that the marching direction must remain supersonic for the marching scheme to be valid. If the Mach number along the marching direction ever drops below one, even though the total Mach number is supersonic, the marching scheme will be unstable. See Shankar and Osher [215] for a derivation of this marching stability condition and additional discussion.

A typical result obtained with the above marching scheme is given in Figs. 22 and 23. The geometry used for this case is a conically cambered wing-body configura-

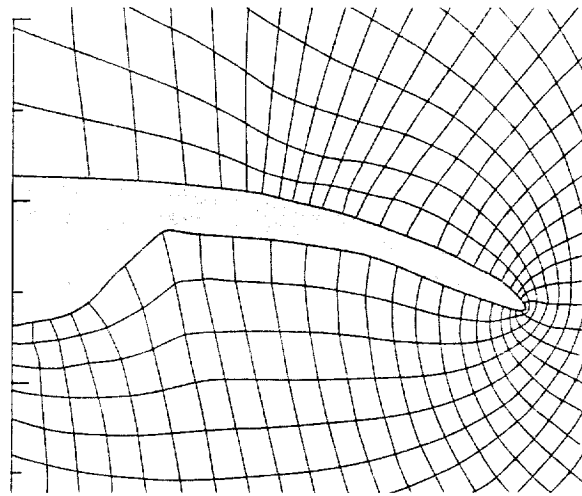


Fig. 22. Cross-sectional grid surrounding a conically cambered wing-body configuration (taken from Shankar and Osher [215]).

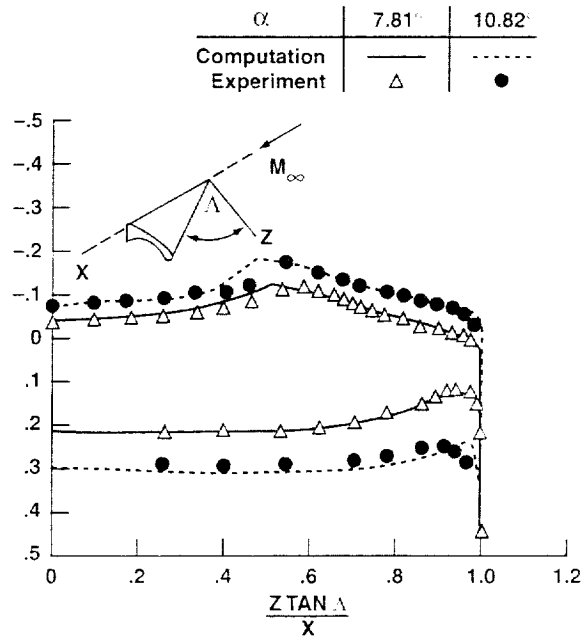


Fig. 23. Surface pressure coefficient comparisons for a conically cambered wing body geometry, $M_\infty = 2$, $\alpha = 7.81^\circ$ and 10.82° . (taken from Shankar and Osher [215]).

tion with a wing leading edge sweep of 57° . The grid for a typical cross-sectional plane (generated using the Steger and Sorenson [216] elliptic grid generation technique) is presented in Fig. 22. Surface pressure comparisons with experiment (taken from Miller et al. [217]) are shown in Fig. 23. The results are presented for $\alpha = 7.81^\circ$ and 10.82° and $M_\infty = 2.0$. Note that the computational-experimental comparisons are excellent.

Another possible characteristic of supersonic marching problems, especially when the freestream Mach number is not significantly above one, is the existence of local “pockets” of subsonic flow. Such a problem can be solved using a hybrid marching-relaxation scheme. The marching scheme is used as long as the marching direction is hyperbolic and a local relaxation scheme is used when the marching direction is elliptic. Such a scheme is presented in Shankar et al. [218]. The key difference between the pure marching scheme presented above and the hybrid scheme is in the handling of the first term in Eq. (52a). For the hybrid scheme this term is simulated using

$$\left(\frac{\rho U}{J}\right) \cong \frac{\theta_i}{\Delta \xi} \left[\left(\frac{\rho U}{J}\right)_{i+1} - \left(\frac{\rho U}{J}\right)_i \right] + \frac{1 - \theta_{i+1}}{\Delta \xi} \left[\left(\frac{\rho U}{J}\right)_{i+2} - \left(\frac{\rho U}{J}\right)_{i+1} \right],$$

where each flux is linearized using the approach described above, and θ is a switching parameter given by

$$\theta = \begin{cases} 1 & \text{if } A_1 - U^2/a^2 < 0, \\ 0 & \text{if } A_1 - U^2/a^2 > 0. \end{cases}$$

When the local marching direction is subsonic the scheme becomes a relaxation scheme, requiring a global iteration over the entire subsonic region of flow. A sonic solution is used as an initial condition for the solution values required at $i + 2$. See Shankar et al. [218] for additional details.

3.9. Time accurate schemes

All numerical schemes reviewed thus far have been designed for steady flow applications. The purpose of this section is to explore numerical schemes designed to solve unsteady problems. There are many unsteady applications in the aerospace field including flutter computations, aircraft maneuver applications and rotorcraft rotor design. Unsteady transonic flow has several interesting aspects that are different from steady transonic flows. In unsteady flow, the motion of a body, e.g., pitching or plunging of an airfoil or wing, strongly affects the resultant aerodynamic forces acting on that body, especially when there are large shock wave excursions. Another unique characteristic is the large phase difference that may exist between the motion of an aerodynamic body and the flow field response to that motion. This is primarily due to the large time-scale variations that exist in such flows. For more discussion of the physical aspects of unsteady transonic flows as well as an early discussion of time-accurate potential equation formulations, the interested reader is referred to Ballhaus and Bridgeman [219]. Additional information on the theoretical and numerical solution aspects of such flows can be found in van der

Vooren and Schippers [220] or Sankar and Malone [201].

Many formulational/algorithmic characteristics that exist for steady potential equation applications also exist for unsteady potential applications. For example, shock wave capture is possible, but only for weak shocks; lifting computations require the utilization of a vortex sheet, across which the velocity potential is discontinuous; and viscous effects, if important, can be added using boundary layer correction techniques. There are other characteristics that are different or have a different emphasis for unsteady applications. For example, unsteady-flow far-field boundary conditions are more sensitive to outer boundary location. Depending on the physics of the unsteady problem being solved and the length of time integration, waves or disturbances can travel outward, reflect back, and corrupt simulation accuracy. A simple method for correcting this problem is to place the outer boundary farther away, as much as an order of magnitude farther than in comparable steady flow problems. Another remedy is to use special far-field boundary conditions. There are two approaches; nonreflecting boundary conditions or analytic far-field boundary conditions. In the first approach, nonreflecting far-field boundary conditions based on characteristic relations (see Engquist and Majda [221,222]) are designed to absorb incoming waves at the far-field boundary. Thus, these waves will not reflect back to contaminate the inner solution. In the second approach, an analytic function describing the far-field unsteady potential solution in response to certain changes in lift is derived and used as a far-field boundary condition (see Fung [223]). Either of these approaches allows the outer boundary to be placed much closer to the inner boundary, saving grid points, computer memory and computer time. Example applications utilizing these two far-field boundary condition approaches to solve unsteady problems can be found in Kwak [224] and Fung [225].

Another characteristic of unsteady potential flow computations is that the circulation, i.e., the potential jump across the wake cut, must be convected downstream in a time-accurate fashion. This characteristic is required to model the unsteady shedding of vorticity downstream of any lifting surface. A relation of the form

$$\Gamma_t + u\Gamma_x = 0,$$

where Γ is the circulation and u is the local streamwise velocity component, is used for this operation. This circulation convection is the major unsteady influence for low-frequency, sub-critical flows. A key approximation in classical potential CFD methods is that Γ is constrained to convect along a coordinate surface. This constraint has historically resulted from transonic fixed-wing applications for which potential CFD methods were originally developed. For these applications the above

convection model is a good simplification and causes little error. See Steger and Caradonna [116] or van der Vooren and Schippers [220] for more discussion on this point and a derivation of the above circulation transport relation.

For time-accurate computations involving a forced unsteadiness, e.g., a wing undergoing a pitch oscillation, flow tangency boundary conditions at the wing surface must take into account movement of the wing. This is accomplished using the boundary condition given by Eq. (32).

One last characteristic important for unsteady algorithms is the use of conservative form. For steady flows conservative form is important for having the correct shock location and strength. For unsteady flows conservative form is also important for having the correct shock location and strength, but also important for having the correct shock speed. A key aspect of this is associated with the numerical algorithm's time-term linearization. Linearization for a steady algorithm is not as difficult as for a time-accurate scheme, because only spatial terms need be considered, and the key issues are spatial accuracy and stability. In addition to these issues, time-accurate schemes must also provide a time-term linearization, which is stable, time accurate, and conservative. The time-term linearization for the full potential equation is difficult because the time term ρ_t is a rather complex function of the dependent variable ϕ . Nevertheless, there are schemes that overcome these difficulties and provide fast and accurate solutions to unsteady aerodynamic problems using potential formulations, as will be seen shortly.

As with the steady flow case, the first successful implementations of unsteady transonic potential solvers are associated with the TSD equation. Early two-dimensional work in this area can be found in Ballhaus and Steger [1], Ballhaus and Goorjian [226], Rizzetta [227], Rizzetta and Chin [228], Houwink and van der Vooren [229], Guruswamy and Yang [230], Edwards et al. [231], Traçi et al. [232] and Isogai [233]. In these efforts, the low-frequency TSD potential equation is solved for a variety of airfoil forced motions, e.g., sinusoidal pitch or plunge oscillation. Three dimensional solutions for the TSD equation followed rapidly and can be found in Caradonna and Isom [234], Borland et al. [235–237], Rizzetta and Borland [238], Chattot [163], Isogai and Suetsugu [239], Guruswamy et al. [240–242], Rodman et al. [243] and Batina et al. [244–247]. In these efforts, several different forms of the three-dimensional unsteady TSD potential equation are solved for a variety of different geometries ranging from isolated wings to nearly complete aircraft. A supersonic freestream capability for both steady and unsteady three-dimensional TSD applications is presented in Gibbons and Batina [248] and Bennett et al. [249]. For supersonic freestream cases, distances to the outer boundary can be reduced and different far-field boundary conditions are required.

Lastly, an unsteady three-dimensional approach that utilizes entropy and vorticity corrections at shock waves (see Section 3.3) is presented in Batina [140]. With these corrections shock wave strengths and positions are in close agreement with Euler equation solvers, even for difficult transonic flow computations.

A major reason for developing an unsteady transonic aerodynamic analysis capability is to be able to predict dynamic aeroelastic characteristics of a wing or airframe. This is particularly important for transonic flow because of the nonlinear behavior of flutter boundaries in this speed regime. Dynamic aeroelastic applications require a time-accurate coupling between the aerodynamics and a suitable structural dynamics algorithm. Example three-dimensional applications for which flutter computations are described include Goorjian and Guruswamy [250] and Guruswamy et al. [251–258] in which the XTRAN3S code is utilized and Bennett et al. [259] and Silva and Bennett [260] in which the CAP-TSD (Computational Aeroelastic Program–Transonic Small Disturbance) code is utilized. A comparison of these two codes is presented in Pitt et al. [261] for a variety of fighter wing aeroelastic computations. The CAP-TSD code is preferred because of its ability to model more complex configurations. The first implementations of unsteady full potential solvers can be found in Isogai [262], Steger and Caradonna [263], Goorjian [117], Malone and Sankar [200] and Shankar et al. [264]. In the first work, the nonconservative form of the full potential equation is used, but with a specially constructed artificial dissipation term that is in divergence form, mimicking a conservative scheme. The latter efforts all utilize conservative form. In all of these cases two-dimensional computations involving simple forced airfoil motions are considered. The first three-dimensional, unsteady full potential studies can be found in Steger and Caradonna [116], Sankar et al. [165,201], Isogai [265], Bridgeman et al. [266], Malone et al. [267] and Shankar and Hiroshi [268]. All of these efforts, except Isogai's work, which uses the previously mentioned nonconservative form with conservative artificial viscosity, solve the conservative form of the full potential equation. Many of these time-accurate schemes have similar characteristics. Thus, it is instructive to look at a particular scheme in more detail.

As mentioned above, a particularly important and difficult aspect of unsteady full potential algorithms is the time linearization, which must be stable and conservative. An example of how such a linearization is constructed is now presented (see Steger and Caradonna [263]). For convenience, only a two-dimensional version is described. Noting the fact that $\rho = \rho(\phi)$, the following Taylor series expansion can be written:

$$\rho = \rho_0 + \left. \frac{\partial \rho}{\partial \phi} \right|_0 (\phi - \phi_0) + O(\phi - \phi_0)^2, \quad (58)$$

where the “0” subscript is used to indicate a “nearby” known state of the solution, e.g., the solution at the previous time step. The derivative of ρ with respect to ϕ is a non-commutable, differential operator derived from a two-dimensional version of Eq. (8) and is given by

$$\frac{\partial \rho}{\partial \phi} = -\rho^{2-\gamma} \left(\frac{\partial}{\partial t} + \phi_x \frac{\partial}{\partial x} + \phi_y \frac{\partial}{\partial y} \right).$$

Substituting Eq. (58) into the time term of the unsteady full potential equation [Eq. (4)] yields

$$\begin{aligned} \frac{\partial}{\partial t} \left[\rho_0^{2-\gamma} \left(\frac{\partial}{\partial t} + \phi_x \frac{\partial}{\partial x} + \phi_y \frac{\partial}{\partial y} \right) \phi \right] &= \frac{\partial}{\partial x} (\rho \phi_x) \\ &+ \frac{\partial}{\partial y} (\rho \phi_y) + \frac{\partial}{\partial t} \left(\rho_0 - \frac{\partial \rho}{\partial \phi} \Big|_{\phi_0} \right) + O(\phi - \phi_0)^2. \end{aligned} \quad (59)$$

Assuming that $\phi - \phi_0$ is small, the error introduced by expanding ρ is second order and therefore, is no larger than that generated by a typical second-order-accurate space-differencing scheme. The above time linearization is conservative and linear in the velocity potential, and thus achieves both of the important goals required of a time linearization.

A complete time-accurate iteration scheme for advancing the velocity potential solution from one time level n to the next $n+1$ that utilizes the above time linearization is given by

$$\begin{aligned} \left[I + h(\phi_{x,i}^n \delta_x + \phi_{y,i}^n \delta_y) - \frac{h^2}{\beta^n} (\bar{\delta}_x \rho_{i+1/2,j}^n \bar{\delta}_x \right. \\ \left. + \bar{\delta}_y \rho_{i,j-1/2}^n \bar{\delta}_y) \right] (\phi_{i,j}^{n+1} - \phi_{i,j}^n) = R_{i,j}^n, \end{aligned} \quad (60)$$

where h is the time step, $\beta = \rho^{2-\gamma}$ and $R_{i,j}^n$ is the n th-iterate residual defined by

$$\begin{aligned} R_{i,j}^n &= \frac{h}{\beta^n} (\rho_{i,j}^n - \rho_{i,j}^{n-1}) + (\phi_{i,j}^n - \phi_{i,j}^{n-1}) \\ &+ \frac{\beta^{n-1}}{\beta^n} (\phi_{i,j}^n - 2\phi_{i,j}^{n-1} + \phi_{i,j}^{n-2}) \\ &+ \frac{h\beta^{n-1}}{\beta^n} (\phi_{x,i}^{n-1} \delta_x + \phi_{y,i}^{n-1} \delta_y) (\phi_{i,j}^n - \phi_{i,j}^{n-1}) \\ &+ \frac{h^2}{\beta^n} (\bar{\delta}_x \bar{\rho}_{i+1/2,j}^n \bar{\delta}_x + \bar{\delta}_y \rho_{i,j-1/2}^n \bar{\delta}_y) \phi_{i,j}^n. \end{aligned} \quad (61)$$

In Eqs. (60) and (61), δ_x and δ_y are standard central first-difference operators and $\bar{\delta}_x$, $\bar{\delta}_y$, $\bar{\delta}_{x^+}$, and $\bar{\delta}_{y^+}$ and standard backward or forward first-difference operators in the x and y directions, respectively. The $\bar{\rho}_{i+1/2,j}^n$ quantity appearing in Eq. (61) is taken from the artificial density approach previously described in Section 3.3 [see Eqs.

(42a) and (42b)]. This term is used to provide the upwind influence required to stabilize any supersonic regions of flow that might appear during the time-accurate iteration process.

In order to avoid costly non-narrow-banded matrix inversions, the iteration scheme given by Eq. (60) is approximately factored using an ADI-type factorization. The resulting iteration scheme is given by

$$\begin{aligned} \left[I + h\phi_{x,i}^n \delta_x - \frac{h^2}{\beta^n} \bar{\delta}_x \rho_{i+1/2,j}^n \bar{\delta}_x \right] \\ \times \left[I + h\phi_{y,i}^n \delta_y - \frac{h^2}{\beta^n} \bar{\delta}_y \rho_{i,j-1/2}^n \bar{\delta}_y \right] (\phi_{i,j}^{n+1} - \phi_{i,j}^n) = R_{i,j}^n, \end{aligned}$$

where the first bracketed term represents a set of tridiagonal matrix inversions along the x direction, and the second bracketed term represents a set of tridiagonal matrix inversions along the y direction. Extension of this scheme to a nonorthogonal, mapped coordinate system involving three spatial dimensions is straightforward and is described in detail in Bridgeman et al. [266].

Recent rotorcraft applications utilizing the conservative full potential equation can be found in Steinhoff and Ramachandran [269], Ramachandran et al. [270,271] and Bridgeman et al. [272] where an unsteady full potential formulation with vortex embedding is used. The fundamental problem with rotorcraft flows is that the shed wake remains in the rotor vicinity for a long time. This determines the spatial and temporal distribution of loads and ultimately the rotor performance and acoustic characteristics. The method of vortex embedding permits such computations because it removes the grid-coordinate convection constraint that is typically associated with time-accurate potential computations. This is accomplished by decomposing the velocity into two components given by

$$\mathbf{q} = \nabla \phi + \mathbf{q}_v. \quad (62)$$

The first part is the standard velocity potential gradient, and the second part is a specified rotational velocity field, \mathbf{q}_v , which contains the shed wake circulation and can be highly localized. This approach is actually a generalization of the standard differencing scheme that is used on any potential-flow wake cut. In the latter case the differencing across the discontinuous wake sheet actually has the same form as Eq. (62). The only difference with vorticity embedding is that the \mathbf{q}_v distribution is 5–6 cells thick and is not constrained to a computational-domain coordinate surface. With the velocity vector definition given by Eq. (62), the full potential equation is given by

$$\rho_t + \nabla \cdot (\rho \nabla \phi) = -\nabla \cdot (\rho \mathbf{q}_v),$$

which is the original full potential equation with an added forcing function. Specification of \mathbf{q}_v can take on

several forms. In Caradonna et al. [273] it is specified as the incompressible velocity field induced by a two-dimensional free vortex. The two-dimensional assumption is valid for this study because the problem being solved is a parallel blade-vortex interaction (BVI) problem, i.e., the wake-vortex core is assumed parallel to the oncoming blade. For more general applications, \mathbf{q}_w can be specified using a thin, well-defined sheet. Nonzero values of \mathbf{q}_w need exist only near the sheet, which is a thickened representation of the rotor wake that serves to produce the proper shed circulation distribution associated with the wake. In this approach, the location of the sheet is computed using a deformable carpet of shed markers. These marker locations can be computed by means of a Lagrangian convection approach, or they may simply be specified using experimental measurement.

Additional recent work utilizing unsteady full potential algorithms can be found in Bridgeman et al. [274,138,145], Chen and Bridgeman [146], Strawn and Tung [275] and Strawn and Caradonna [276]. In these references three-dimensional transonic flow computations about rotorcraft rotors are described using a conservative full potential approach. The first four references

include viscous effects and a nonisentropic shock wave correction designed to more closely model the Euler equations (see Section 3.3). The viscous effects are included using one of two options: a two-dimensional momentum integral method that uses a “strip” approach or the three-dimensional finite-difference boundary layer equation approach of Van Dalsem and Steger [277].

Typical unsteady rotor results from Bridgeman et al. [138] showing surface pressure comparisons for both the isentropic and nonisentropic approaches are presented in Fig. 24. The experimental data are from the Army 7 × 10 Tunnel at Ames Research Center. The results are for an untwisted, rectangular, NACA 0012 rotor blade with an aspect ratio of 7.125, an advance ratio (μ) of 0.246 and a tip Mach number (M_T) of 0.763. The advancing rotor blade solution is displayed at a fixed radius near the tip ($r/R = 0.876$) for six different azimuthal stations ($\psi = 30, 60, 90, 120, 150, \text{ and } 180^\circ$). As the blade advances a shock wave forms, grows in strength and finally disappears at $\psi = 180^\circ$. Generally, both computations are in good agreement with experiment. However, the nonisentropic result moves the shock wave upstream, in better agreement with experiment, especially at the stations where the shock is strongest.

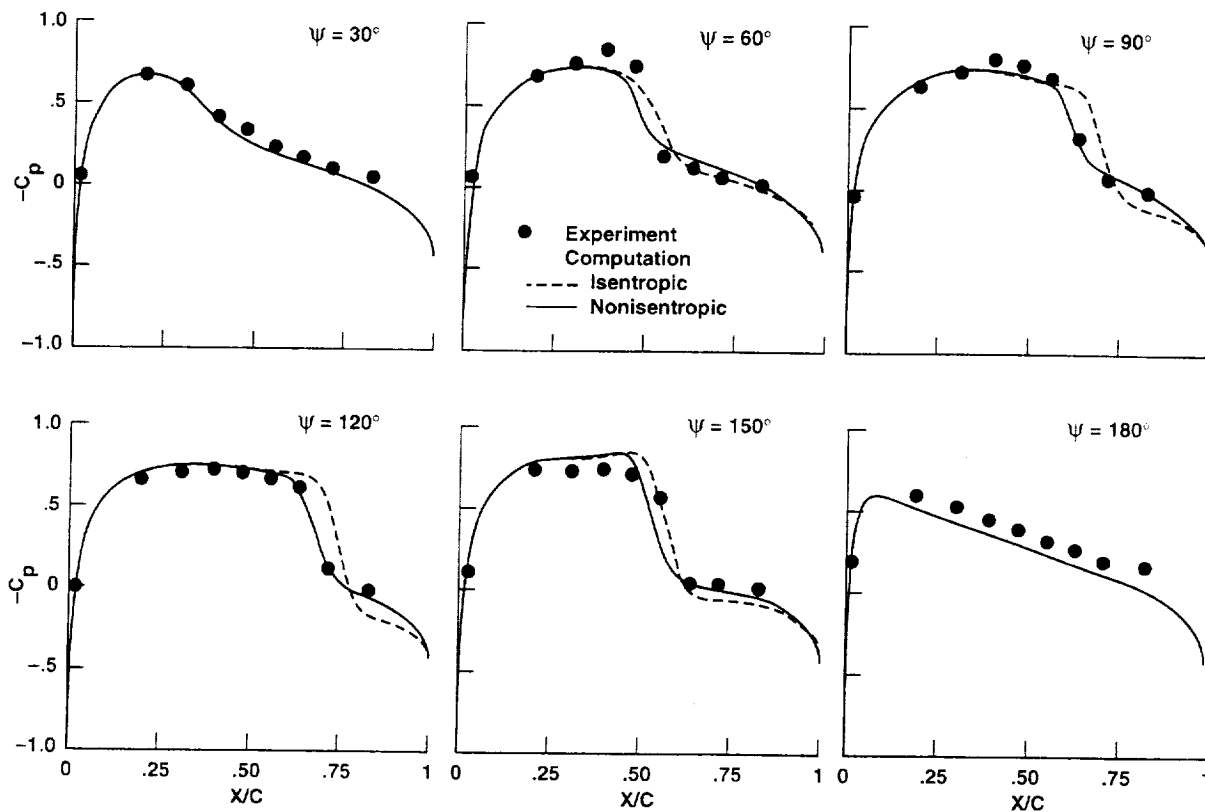


Fig. 24. Surface pressure comparisons for an untwisted, NACA 0012 rotor blade at six different azimuthal angles, $AR = 7.125$, $\mu = 0.246$, $M_T = 0.763$, $r/R = 0.876$, taken from Bridgeman et al. [138].

3.10. Design methods

So far only analysis methods utilizing nonlinear potential formulations have been examined. In analysis the geometry, the freestream flow conditions, and all boundary conditions are completely specified and the goal is to obtain the flow field. In design the situation is more complex because of the large variety of approaches that are available. In one approach the surface pressure distribution is specified, and the goal is to obtain the geometry that produces this pressure. Still other design approaches seek geometrical changes that produce optimal aerodynamic performance, e.g., minimum drag-to-lift ratio. The whole field of aerodynamic design and optimization is complex because each approach may have variations based on the applicable speed regime, the governing equation formulation being used, and/or the analysis method being used. The purpose of this section is to present a brief survey of design methods that utilize a nonlinear potential governing equation approach for transonic aerodynamic design. For the more general topic of aerodynamic design and optimization, the interested reader is referred to AGARD [278] or Dulikravich [279,280].

3.10.1. Indirect methods

The first design method class to be discussed is called the indirect method. In this approach the designer does not have precise control over either the geometry or the solution. One example of an indirect method is the hodograph approach, which involves transforming the full potential equation such that the independent variables become the velocity components. In the hodograph plane the governing equation is linear, and thus, solutions can be constructed using the powerful idea of superposition. However, transformation back into the physical plane can lead to difficulties as some solutions may not have physical meaning. An additional drawback of the hodograph method is that only shock-free solutions in two dimensions can be obtained. The hodograph method has not been used widely in recent years and will not be discussed further. For more information on this approach the interested reader is referred to Boerstoe [281] and Bauer et al. [71].

Another example of an indirect design method is the fictitious gas approach first devised by Sobieczky et al. [282]. In this approach the governing potential equation is modified in the supersonic flow regime so as to retain an “elliptic” nature over the entire “transonic” flow domain. This may be accomplished in a number of ways providing the scheme maintains both local and global conservation of mass. The simplest approach is to set the local density to the critical value of density ρ^* whenever the flow becomes supersonic. Thus, assuming an approach based on the two-dimensional conservative full potential equation, the supersonic flow regime is solved

using

$$\left(\frac{\rho^* U}{J}\right)_{\xi} + \left(\frac{\rho^* V}{J}\right)_{\eta} = 0 \Rightarrow \left(\frac{U}{J}\right)_{\xi} + \left(\frac{V}{J}\right)_{\eta} = 0,$$

where the latter equation is the inherently elliptic Laplace equation mapped to general coordinates. In subsonic regions the traditional full potential computational procedure is unchanged. With this approach the entire domain is elliptic, and solutions obtained will be shock-free. Once a solution with the fictitious gas model is obtained, the sonic line information is saved.

The next step utilizes the sonic line as initial data and a hyperbolic marching scheme to generate a physically valid solution in the supersonic region of flow. In two dimensions this scheme is typically the well-known method of characteristics, e.g., see Dulikravich and Sobieczky [283]. In three dimensions a more general marching scheme based on the full potential equation is utilized, e.g., see Yu [284]. Once this solution is generated the original aerodynamic shape wetted by supersonic flow will no longer be a stream surface in the new flow solution. Thus, a new aerodynamic shape in this region must be computed by finding the stream surface that connects the upstream and downstream sonic points at each wing station. Finally, the last step in this approach consists of testing the modified aerodynamic shape using the original method in analysis mode. The resulting solution should be shock free or nearly shock free. The fictitious gas method has been used for a large number of applications including airfoil and wing design by Sobieczky et al. [282], cascade design by Dulikravich and Sobieczky [283] and wing design by Yu [284], Fung et al. [285] and Raj et al. [286].

There are two difficulties with the fictitious gas approach. The first is that shock-free designs are not always possible for all combinations of initial geometry and freestream flow conditions. If such a set of conditions has been chosen the supersonic marching solution may not converge properly. The second difficulty is that the marching problem for the supersonic flow domain is not well posed in three dimensions, i.e., small changes in the initial data can produce large changes in the final solution. This difficulty is most severe for small aspect-ratio wings involving large gradients in the spanwise direction. See Fung et al. [285] for more discussion on these latter two points.

3.10.2. Inverse methods

The inverse method in aerodynamic design seeks to determine the aerodynamic shape for a specified surface pressure distribution, i.e., the “inverse” of the normal analysis approach. Sometimes this design approach is called the surface design method. An advantage of this approach is that it offers direct control over aerodynamic forces and moments (through specification of the surface

pressure). In addition, by utilizing proper constraints on the adverse pressure gradient, a degree of control on boundary layer separation is also available, even for inviscid implementations. The biggest difficulty of the inverse design method is selecting a pressure distribution that will achieve the best aerodynamic performance for a given set of constraints. Clearly, experience helps in this area, but knowing what is good aerodynamically and (at the same time) does not over constrain the inverse method and prevent convergence can be a problem.

The inverse design method was first implemented using a two-dimensional TSD approach by Steger and Klineberg [287] and Langley [288] and a two-dimensional full potential approach by Tranen [289], Carlson [72], Volpe and Melnik [290] and Volpe [291]. Subsequently, extensions to three dimensions have been made by Shankar et al. [292,293] for the TSD equation, by Henne [294] and Garabedian and McFadden [295] for the nonconservative full potential equation, and by Shankar [385] for a conservative full potential approach. A good discussion of early inverse methods for transonic airfoil and wing design, comparing and contrasting various characteristics is given in Slooff [296]. More recent applications for the inverse design approach include Gally and Carlson [297] and Ratcliff and Carlson [298] where the TAWFIVE analysis code of Street [97] is modified for the wing inverse design problem, Takanashi [299] where a wing design method based on a residual-correction concept is presented, Carlson and Weed [300] where a wing design method is developed using a Cartesian-like grid system, Malone et al. [301] where a method is applied to transonic nacelle design, Hassan and Charles [302] where a method is presented for helicopter rotor design, and de Mattos and Wagner [173] and de Mattos [172] where a method is used for both wing and wing/fuselage design.

Although implementation details vary from approach to approach, the basic inverse design method has several common steps. To gain insight into this class of design methods, details from the implementation of de Mattos and Wagner [173] are now described. For brevity only an airfoil algorithm is presented using the two-dimensional conservative full potential governing equation given by Eqs. (37). For this presentation the ξ and η coordinates are assumed to be along and away from the airfoil surface, respectively. The first step in the inverse design procedure is to generate an analysis solution using the initial geometry. Second, a modified velocity potential along the airfoil surface is computed using the specified pressure distribution. This is accomplished by converting the specified pressure into a specified speed using

$$C_p|_{\text{specified}} = \frac{2}{\gamma M_x^2} \left\{ \left[1 + \frac{\gamma+1}{2} (M_x^2 - q_{\text{specified}}^2) \right]^{1/(\gamma-1)} - 1 \right\},$$

which is derived from the density-speed relation, the speed of sound definition and the steady Bernoulli equation. The specified speed can be related to the velocity potential using its general coordinate definition

$$q_{\text{specified}}^2 = U^n \phi_\xi^n + V^n \phi_\eta^n,$$

where the n superscript is used to indicate the n th design iteration. In order to achieve the specified surface pressure, it is assumed that a perturbation in the velocity potential $\Delta\phi$ is required. To this end the above relation can be written as

$$q_{\text{specified}}^2 = (U^n + \Delta U^n)(\phi_\xi^n + \Delta\phi_\xi^n) + (V^n + \Delta V^n)(\phi_\eta^n + \Delta\phi_\eta^n),$$

which after dropping higher-order perturbations and applying the surface condition $V = 0$ becomes

$$\Delta\phi_\xi^n = \phi_\xi^{n+1} - \phi_\xi^n = \frac{q_{\text{specified}}^2 - (q^n)^2}{2U^n}.$$

Using this relation and the assumption that the leading edge value of the velocity potential does not change, modified surface velocity potential values downstream of the leading edge can be computed.

The third step in the inverse design approach is to compute a new global solution with a new airfoil surface boundary condition. That is, instead of solving a Neumann problem involving flow tangency at the airfoil surface, a Dirichlet problem is solved. The newly computed velocity potential is used as the Dirichlet boundary condition.

In general, the new global solution will have a nonzero value of V at the airfoil surface. The fourth inverse design step is to use this nonzero value of V to compute a change in the airfoil geometry. A formula to achieve this can be derived from $dy/dx = \phi_y/\phi_x$. This reestablishes flow tangency at the airfoil surface. Once the airfoil shape has been changed the whole process starts over with step one and continues until a suitable level of convergence has been achieved. As pointed out in de Mattos and Wagner [173], it is important to underrelax the shape changes that are implemented in this algorithm using a relation of the form

$$y^{n+1} = \omega \overline{y^{n+1}} + (1 - \omega)y^n,$$

where $\overline{y^{n+1}}$ is the initial updated value of y before underrelaxation, y^{n+1} is the final value of y used to update the airfoil shape, and ω is the relaxation factor that must be below one.

One last point regarding this technique is in order. It has to do with trailing edge closure. In general, if this issue is not addressed, the above described inverse design approach will produce an airfoil with an open trailing

edge, or even worse, a trailing edge with negative thickness. To keep this from happening a function $\delta(x)$ is added to the airfoil y -coordinate distribution after each design iteration. This function is given by

$$\delta(x) = -\Delta_{te} \left(\frac{x}{c} \right),$$

where the quantity Δ_{te} is the airfoil trailing edge thickness and x/c is the normalized distance along the airfoil chord. The problem of trailing edge closure is a common one among inverse design methods and must be addressed in order for sensible results to be obtained.

An example result from an inverse wing design method taken from Gally and Carlson [297] is presented in Fig. 25 for an initial wing with NACA 0012 airfoil sections. The initial pressure distributions, user-specified target pressures and the final design pressures are presented for two different wing span stations, 30 and 70% of semi-span. At both stations the new pressures are in close agreement with the target pressures. The upper-surface shock has been eliminated or considerably weakened at both stations.

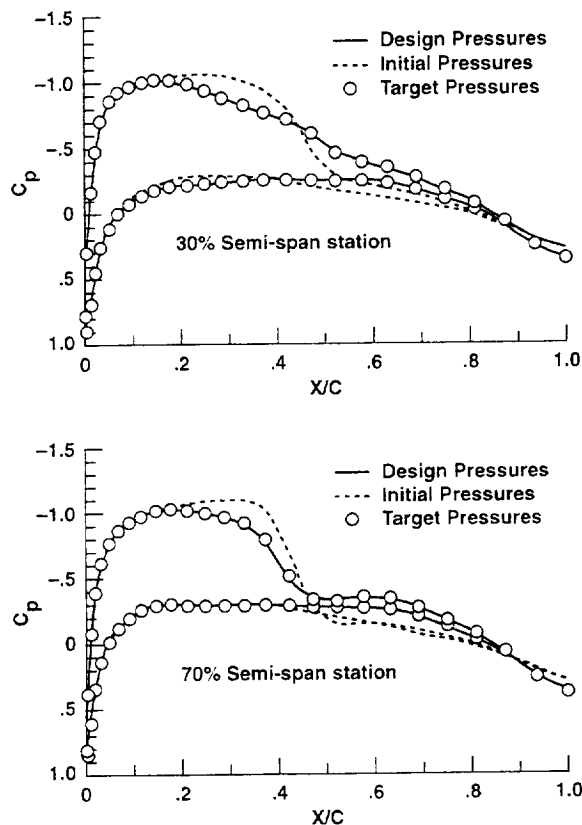


Fig. 25. Comparison of initial, target and final design pressures using an inverse transonic wing design procedure, taken from Gally and Carlson [297]. (a) 30% semi-span station. (b) 70% semi-span station.

3.10.3. Numerical optimization design (gradient methods)

Numerical optimization using gradient-based methods (GM) in aerodynamic design has received much attention in recent years. The reliability and success of gradient methods is based on smoothness of the design space and the existence of only a single global extremum. A good review of gradient methods used in aerodynamic design is presented by Reuther [303]. The general idea associated with this broad class of methods consists of the following steps. First, determine the optimization's objective, e.g., minimization of the drag-to-lift ratio, minimization of the least-squares error between the actual and a prescribed pressure distribution, etc. Second, the geometry to be optimized must be parameterized. This parameterization must completely describe the geometry (or the portion that is to be optimized) and must lend itself to discrete variations that can be independently modified. In many (but not all) gradient method implementations it is advantageous to parameterize the geometry with the minimum number of parameters that will still completely describe the applicable design space. Examples of aerodynamic shape parameterizations are given in Hicks and Henne [304] where a series of "bump" functions is used or Burgreen and Baysal [305] where a series of B-spline control points is used.

Mathematically, the dependence between the objective f and the decision variables resulting from the design space parameterization \mathbf{x} can be expressed as follows:

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_k, r, \dots, x_K),$$

where K is the total number of decision variables used in the design space discretization. In addition, problem constraints must be identified, e.g., minimum wing thickness, minimum wing volume, etc. These can be included in the objective function as a penalty or included as separate inequality constraints. Different gradient methods allow the inclusion of constraints in different ways.

The third step is to compute the direction in the design space (away from a specified initial condition) that minimizes the objective. This is achieved by using (for example)

$$\mathbf{x}^{n+1} = \mathbf{x}^n - s^n \nabla f(\mathbf{x}^n)^T, \quad (63)$$

where \mathbf{x}^n is the decision variable vector from the previous iteration, \mathbf{x}^{n+1} is the improved decision variable vector, s^n is the n th-level step-size vector, and $\nabla f(\mathbf{x}^n)^T$ is the n th-level column vector of sensitivity derivatives. Eq. (63) represents the simplest type of gradient method, often called the steepest decent method. Many others are available including conjugate gradient, Newton or quasi-Newton approaches. The interested reader is referred to Luenberger [306] or Reuther [303] for general details in this area. Additional comments regarding specific optimization packages that have been used in aerodynamic optimization can be found in Vanderplaats [307] for

CONMIN (constrained-function minimization), Gill and Murray [308] for QNMDIF (quasi-Newton method with difference approximations for the derivatives), Gill et al. [309] for NPSOL (nonlinear programming solver), Vanderplaats [310] for ADS (automated design synthesis) and Cheung [311] for IOWA (parallel optimization with aerodynamics). The technique for determining sensitivity derivatives is a key item in any gradient-based design approach and has received much attention. The simplest approach, often called the “brute-force” or finite-difference method, consists of using finite-difference formulas to compute sensitivities of the form

$$\frac{\partial f}{\partial x_k} \cong \frac{f(x_1, x_2, \dots, x_k + \epsilon_k, \dots, x_k) - f(x_1, x_2, \dots, x_k, \dots, x_k)}{\epsilon_k},$$

where ϵ_k are the user-specified difference intervals. Both values of f used in the above equations numerator are computed using a separate CFD simulation from an appropriate analysis code. Care must be taken in this approach to make sure that each solution is adequately converged, usually with tighter convergence than in an analysis computation. If not, the lack-of-convergence error can cause large errors in sensitivity derivative computation and difficulties in the optimization process. If a design problem using the above approach has K decision variables, the sensitivity derivative computations for each design iteration will require $K + 1$ function evaluations. Specifically, $K + 1$ complete CFD analysis solutions must be computed, one solution for the unperturbed or baseline geometry and K solutions corresponding to the K perturbed geometries. After the sensitivity derivatives have been computed and the steepest descent direction is determined, a “line search” along this direction is required to determine values for the step size vector s . One approach for this operation is to compute several values of f using different step sizes and then compute the minimum value of f using a curve fitting procedure (see Cheung [312] or Reuther [303]). Other line search procedures utilized for aerodynamic optimization are described in Melvin et al. [313].

Overall, gradient-based methods for typical aerodynamic optimization problems require from several iterations to several tens of iterations to converge (depending on the method used and the number of decision variables). Thus, the total number of CFD analysis solutions required for this type of optimization approach can easily number in the thousands. Because nonlinear potential methods require little computer time per solution relative to Euler or Navier–Stokes approaches, a potential-based finite-difference optimization approach may provide suitable turnaround times for the design environment whereas a similar approach based on the Euler/Navier–Stokes equations would be too expensive. Examples utilizing the finite-difference gradient optimization

method in conjunction with nonlinear potential solvers include Hicks et al. [314], Kennelly [315] and Ghielmi et al. [316] for airfoils; Haney et al. [317], Hicks [318], George et al. [319] and Cosentino and Holst [170] for wings; Destarac et al. [320] for wings and wings with propulsion effects; Reneaux and Allongue [321] for helicopter rotors; Cheung and Holst [171] for wing-body applications; and Aidala [322] for wing-body-canard configurations.

Other methods for evaluating sensitivity derivatives that seek to reduce the large computational cost associated with the finite-difference method are the quasi-analytic (QA) method of El-banna and Carlson [323,324] and Arslan and Carlson [325] and the method based on control theory presented by Jameson [326,327]. In the QA approach the sensitivity derivatives are obtained by solving large sparse systems of matrix equations. The elements of these matrix equations are developed by taking analytic derivatives of the numerical or discrete governing equations (with the aid of a symbolic manipulation program). In this approach an entire set of sensitivity derivatives is obtained with the solution of a single matrix equation. Sensitivity derivatives obtained using this approach are in good agreement with those obtained from the finite-difference approach. More work needs to be completed in this promising area to assess this method's abilities in actual design optimization applications.

The second method involves the numerical solution of an adjoint equation derived using control theory (see Reuther [303] for a derivation of the adjoint equation for the full potential equation in two dimensions). The optimization method using the adjoint approach has the following steps: First, solve for the flow field using a typical analysis method. Next, solve the adjoint equation for the sensitivity derivatives. Using these sensitivity derivatives and a suitable gradient optimizer, obtain an improved design with updated decision variables. Finally, repeat all steps until a sufficient level of convergence is achieved. This approach is superior to the finite-difference approach for generating sensitivities because all the sensitivities are obtained (no matter how many there are) by solving one adjoint equation, with a cost on the order of one flow field solution. Thus, for design problems containing a large number of decision variables, the cost savings for this approach over the finite-difference approach is significant. Because any number of decision variables can be used in the adjoint approach without significantly increasing the sensitivity derivative computational cost, it is natural to utilize very large numbers of decision variables in an attempt to improve optimization results. The cases that are reported in Jameson [326,327] follow this strategy by utilizing every surface grid point as a decision variable. In the case of wing optimization, this produces as many as several thousand decision variables. As described in Reuther [303]

this increased design space resolution creates several difficulties. First of all, the large number of decision variables can introduce high frequencies into the flow solution causing difficulties for the flow and adjoint solvers. The high-frequency aspects of the design space can cause local extrema to appear, which make proper convergence of the gradient optimizer difficult. In the work of Jameson [326,327] a gradient smoother is introduced to solve this problem by eliminating (or smoothing) high-frequency details. Although this approach works, it eliminates (at least some of) the apparent advantage of being able to efficiently handle a large number of decision variables. It also introduces an undesirable aspect: the dependence of the optimization algorithm on a smoothing process. In contrast, the adjoint-based gradient optimization method of Reuther [303] utilizes a smaller number of decision variables following the design space parameterization of Hicks and Henne [304] and produces good results without a gradient smoother. A typical result using a gradient optimization algorithm is presented in Section 3.11.3.

3.10.4. Numerical optimization design (genetic algorithms)

The last optimization method discussed in this section is called the genetic algorithm (GA) approach (or sometimes the random search approach). The basic idea associated with this approach is to search for optimal solutions using the theory of evolution. During solution iteration (or “evolution” using GA terminology) the decision variables are manipulated using various operators (selection, combination, crossover, mutation) to create new design populations, i.e., new sets of decision variables. General details of such genetic algorithms and the specific operators used in them can be found in Goldberg [328], Schwefel [329] and Davis [330]. Each design is evaluated using an objective-like “biological fitness function” to determine survivability. Constraints can easily be included in this approach. If a design violates a constraint, its fitness function is set to zero, i.e., it does not survive to the next evolution level. Because GA optimization is not a gradient-based optimization technique, it does not need sensitivity derivatives. It theoretically will work well in non-smooth design spaces. The ability to arbitrarily mutate allows a GA optimization approach (theoretically) to find the global extrema in design spaces containing several or perhaps many local extrema. A disadvantage of the GA approach is expense. In general, the number of function evaluations required for a GA algorithm exceeds the number required by a finite-difference-based gradient optimization. Example applications utilizing potential-based flow solvers in the context of GA optimization can be found in Quagliarella and Della Cioppa [331] for airfoil applications, Vicini and Quagliarella [332] for multi-point and multi-objective airfoil applications and Obayashi et al. [333] for multi-disciplinary optimization of transonic wings.

Two simple optimization examples involving two-dimensional linear aerodynamics that compare the GA approach with a typical gradient-based optimization method (GM) are now discussed. The first comparison (taken from Obayashi and Tsukahara [334]) is for the optimal design of a subsonic airfoil using a linear panel method to evaluate the aerodynamic “fitness” of each design variation. The lift is maximized under airfoil thickness and angle of attack constraints. Linear combinations of four existing airfoils are used to establish the design space. This produces a very general but “noisy” design space. For this case the GA and GM algorithms produce optimized lift coefficient values of 2.48 and 1.716, respectively. The GA and GM algorithms require 972 and 159 function evaluations, respectively. The GM algorithm results are from the best of four separate runs that each utilized different initial conditions.

The second example (taken from Bock [335]) is for the optimal design of a symmetric, sharp-edged airfoil at zero lift in supersonic flow. The wave drag is minimized at a fixed freestream Mach number of 1.732 under a thickness constraint. The aerodynamic “fitness” is evaluated numerically using shock-expansion theory. The airfoil shape is represented as the superposition of a series of five Legendre polynomials plus a triangle function, yielding a total of six decision variables. For this case the GA and GM algorithms converge to drag coefficient values of 0.0308 and 0.0281, respectively. The theoretical optimal value of the wave drag for this case is 0.0279. The GA and GM algorithms require about 300 and 60 function evaluations, respectively. The design space in this case is much smoother than in the previous case and is (apparently) a key reason for the superior results produced by the GM algorithm. In addition, (as stated by the author) lack of convergence of the GA scheme to the theoretical value of minimum drag may be due to some inappropriate aspect of the mutation algorithm that is utilized.

Based on the above results, the following observations can be made. The GA approach is computationally expensive, requiring 5–6 times the number of function evaluations required by the (already expensive) GM approach. The GA approach, which demonstrates tremendous results in the first example, especially in view of the noisy design space, is a disappointment in the second. This suggests more work is required to properly evaluate this approach for aerodynamic design. In particular, characteristics of the GA approach for more realistic nonlinear design problems need to be developed.

3.11. Methods developed for complex geometry applications

The purpose of this section is to review transonic full potential methods that have been developed for complex geometry applications including complete or nearly complete aircraft. This area can be divided into four major

sub-areas according to the flow field discretization approach: chimera zonal grids (sometimes called overset grids), patched zonal grids, Cartesian unstructured grids and unstructured grids. Each of these discretization types is sketched in Fig. 26 for a simple two-dimensional body inside a rectangular domain. The chimera zonal grid approach utilizes two or more grid zones that are generated separately and overlap in a general fashion. The flow field solutions in each chimera grid are connected during flow solver iteration using a general interpolation scheme (Fig. 26a). The patched zonal grid method typically utilizes several to many separate grid zones that interface along common boundaries (Fig. 26b). Each of the individual grids in the chimera and patched zonal grid approaches typically utilizes a structured grid composed of quadrilaterals in two dimensions or hexahedrons in three dimensions. The unstructured Cartesian-grid approach utilizes a grid composed of squares in two-dimensions or cubes in three dimensions. Each Cartesian grid cell can be discontinuously subdivided into smaller cells in regions of high flow gradient (Fig. 26c). The unstructured grid approach typically utilizes flow-domain discretizations composed of triangles in two dimensions or tetrahedra in three dimensions (Fig. 26d).

For the unstructured grid approaches the flow solver is typically a finite-volume or finite-element method

(FEM). For the zonal grid approaches the flow solver is typically a finite-difference or finite-volume method. Since most of the methods highlighted in previous sections of this review have been in the latter category, the FEM approach (or more generally, unstructured grid methods) will be primarily highlighted in this section. Selected complex geometry results are presented to allow a complete evaluation of the full potential approach in aircraft analysis and design.

3.11.1. Zonal grid methods

For the purpose of this review, zonal grid methodology has been organized into two categories, chimera and patched methods. The primary zonal grid method utilized in CFD applications is the chimera grid approach, which will be emphasized in this section. The zonal grid approach (in general) and the chimera zonal grid approach (in particular) are versatile techniques for obtaining aerodynamic results for complex configurations including reasonably complete aircraft. In the chimera zonal grid approach a separate boundary conforming grid is generated about each major feature of a complex aerodynamic configuration. For example, for a transport aircraft consisting of a wing/body/pylon/nacelle, a total of five grid zones might be used, one for the near field surrounding each of the major geometric features (wing,

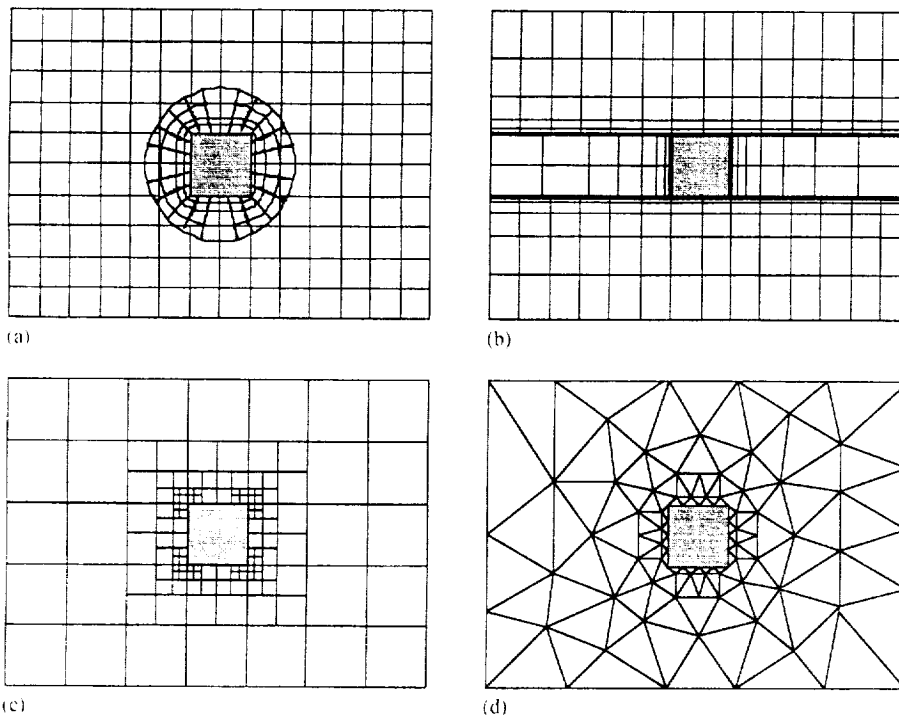


Fig. 26. Sketch of the different types of flow field discretization schemes that are designed to enhance simulations for complex geometries. (a) Chimera or overset zonal grid approach. (b) Patched zonal grid approach. (c) Unstructured Cartesian grid approach. (d) Unstructured grid approach.

body, pylon, and nacelle) and a fifth background grid to “connect” the near field grid zones to freestream. Each of the component grids is generated without regard to any of the other component grids. Each boundary grid point receives its boundary condition information from either freestream, flow tangency (assuming the solver is inviscid), symmetry or from another component grid using interpolation. Some of the grid cells generated for one component of the geometry may have grid points that lie inside other components of the geometry. Computations at such points are handled by an “IBLANK array multiplier”, which has a value associated with each grid point. IBLANK is equal to one for points in valid flow regions (called field points) and equal to zero for points in invalid or “blanked out” flow regions. Blanked-out grid points that lie immediately adjacent to field points are called fringe points or sometimes intergrid boundary points (IGBPs). Fringe points require interpolated information from a neighboring grid zone every iteration. Blanked out points that are not immediately adjacent to field points are called hole points. Computations proceed in an identical fashion at all grid points, which permits efficient code vectorization and/or parallelization, but because of the IBLANK array multiplication, only grid points in valid regions of flow (i.e., field and fringe points) get updated as the iteration proceeds.

Early work in the development of the multi-zone approach for transonic potential calculations can be attributed to the grid embedding approach used for solving the TSD potential equation, e.g., see Boppe [66], Boppe and Stern [67] and Shankar and Malmuth [64]. In these references the TSD equation is solved for the flow about various three-dimensional configurations including a reasonably complete aircraft. Early work in the development of the multi-zone overset approach applied to numerical solution of the full potential equation can be found in Atta [336] for airfoils, Lê [337] for wing/body geometries and Atta and Vadyak [338] for wing/nacelle configurations. Formalization of the chimera approach with the IBLANK array logic can be attributed to Steger et al. [339], Benek et al. [340] and Dougherty et al. [341]. In these references the scheme was first given the name “chimera” and basic concepts of the approach were developed. Applications consisted of relatively simple two- and three-dimensional simulations, primarily involving the Euler equations.

The chimera and patched zonal grid approaches have been further developed for solving the full potential equation by Ecer and Spyropoulos [118] for wing-body combinations; by Epstein et al. [342] for nearly complete aircraft configurations; by Lifshitz et al. [343,344] for airfoils inside wind tunnel walls with viscous effects; by Holst [345,122,346] for a variety of three-dimensional applications including a wing/body/nacelle; and by Sankar et al. [347], Bangalore et al. [348], Berkman et al. [349] and Moulton et al. [350,351] for a variety of hybrid

applications. In the latter area the term “hybrid” refers to the use of different flow solvers in different grid zones. This is a particularly interesting aspect of the zonal grid approach that demonstrates a significant amount of flexibility. Typically, the flow field in an inner grid zone, e.g., next to the geometry’s surface, is solved using a Navier–Stokes approach. The flow field in an outer grid zone is solved using a full potential solver. The detailed viscous flow field physics associated with shock-wave/boundary-layer interaction or dynamic stall is handled with the Navier–Stokes solver and the relatively simple outer flow region is handled using the computationally efficient full potential approach. A factor of two reduction in CPU time with no degradation in solution accuracy relative to a fully Navier–Stokes approach is reported in Sankar et al. [347].

An example result (taken from Holst [346]) using a chimera grid approach to compute the transonic flow about a wing-body-nacelle geometry is shown in Figs. 27 and 28. The chimera grid for this geometry consists of five grid zones, including a wing, a fuselage and an outer Cartesian-like grid. In addition, there are two nacelle grids, a body-conforming grid surrounding the nacelle surface and a stretched Cartesian grid that surrounds the inner nacelle grid. A hyperbolic grid generation code (see Steger and Rizk [352] or Chan et al. [353]) is used to construct the wing, fuselage and inner nacelle grids, while a simple algebraic grid generator is used to construct the other two Cartesian-like grids. A cut approximately through the vertical nacelle symmetry plane showing the wing grid and the two nacelle grids is displayed in Fig. 27. Fig. 28 shows Mach number contours on the wing upper surface, on the fuselage in the vicinity of the wing/fuselage intersection, and on the upper and outboard portions of the nacelle. The results are computed at $M_\infty = 0.9$, $\alpha = 2^\circ$ using a moderate-sized grid consisting of 689,591 total points. The shock wave structure of this solution on the wing, fuselage and nacelle surfaces is clearly visible. A shock extends across the entire aft portion of the upper wing surface from the tip to the root and continues around the upper fuselage. The interference shock caused by the close proximity of the wing and the nacelle is clearly visible on the aft nacelle surface. This computation required about 9 min of CPU time on a high-end workstation running at about 70 MFLOPS.

3.11.2. Unstructured grid methods

Examples of early unstructured grid FEM approaches used to solve the full potential equation for transonic flow are presented in Glowinski et al. [354], Ecer and Akay [355] and Vigneron et al. [356]. A particularly interesting FEM issue for transonic potential flow solutions is how to stabilize supersonic regions of flow. An artificial dissipation term that leads to an upwind influence is typically used. Deconinck and Hirsch [111,184], Akay and Ecer [110] and Eberle [114,115] all use a form

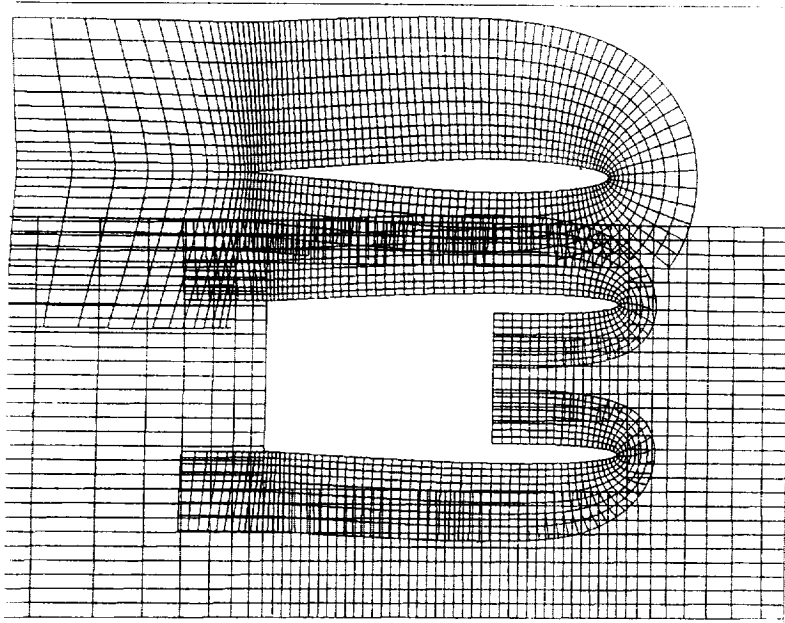


Fig. 27. Selected grid surfaces from the wing/body nacelle chimera grid used in the Fig. 28 computation (nacelle vertical plane of symmetry), taken from Holst [346].

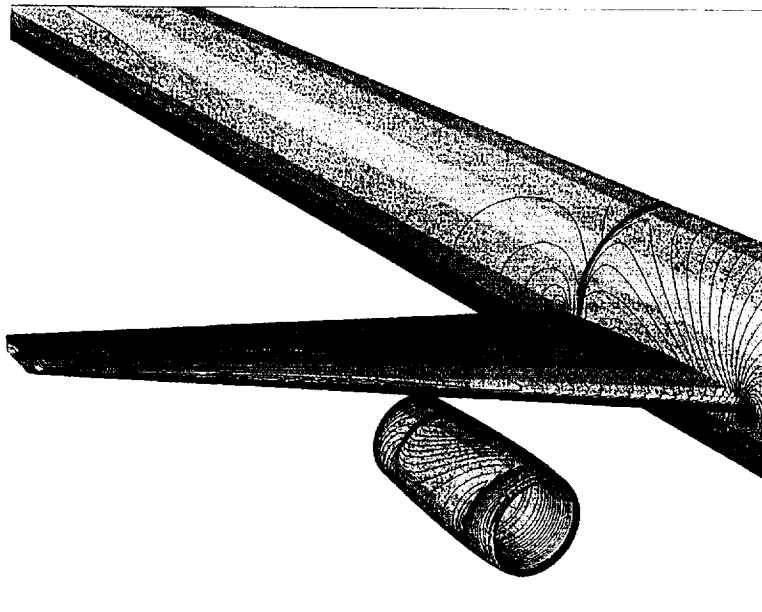


Fig. 28. Mach number contours showing the computed solution on selected surfaces of the wing-body-nacelle configuration, $M_\infty = 0.9$, $\alpha = 2^\circ$, taken from Holst [346].

of the artificial density scheme presented in Section 3.3. Results from a variety of artificial density schemes applied in a finite element context are presented and compared in Habashi and Hafez [132]. Applications in these studies range from airfoils and cascades to wings. Still another supersonic flow stabilization routine for trans-

onic flow applications is presented and demonstrated in Bristeau et al. [203,357] and Periaux [358]. In this approach a least-squares conjugate gradient method is used to solve the finite element equations. The least-squares functional is modified to include a penalty function that becomes very large for expansion shocks without

becoming large for normal compression shocks. This effectively takes on the role of an artificial dissipation term in stabilizing supersonic regions of flow. Of all early studies in the area of full potential FEM applications to transonic flow, the work of Heckman [359] is of particular note. In this study a transonic flow simulation about a nearly complete business jet consisting of a wing/body/nacelle is presented. This demonstrates the capabilities of the FEM method and the unstructured grid approach for handling complex geometries.

Recent studies exploring the theoretical aspects of the FEM approach are presented in Wong and Hafez [205], Glowinski [360] and Berger et al. [361]. Other recent applications of unstructured grid methods for solving the full potential equation include Whitehead and Newton [362] for cascades, Bristeau et al. [363] for a variety of applications, Mehta and Jayachandran [364] for axisymmetric bodies, Kinney et al. [365] and Kinney and Hafez [139] for wings, and Kinney et al. [366–368] for nearly complete aircraft.

To gain more insight into the characteristics of the FEM approach for solving the full potential equation, additional details following the recent work of Kinney et al. [365] are now presented. The governing equation utilized for this implementation is written in steady Cartesian coordinates [Eq. (9)], and the density relation uses $\rho_x \cdot q$, nondimensionalization. Kinney et al. [365] utilize a flux-based upwind procedure to stabilize supersonic regions of flow in the context of a general unstructured grid approach. Following the approach of Osher et al. [129] an elemental flux is defined using

$$\overline{\rho q} = \begin{cases} 0 & \text{if } M < 1, \\ \rho q - \rho^* q^* & \text{if } M \geq 1, \end{cases}$$

where ρ^* and q^* are sonic values of the density and fluid speed, respectively. With this flux definition the density is upwinded using

$$\hat{\rho} = \rho - \frac{\Delta \mathbf{s}}{q} \frac{\hat{c}}{\hat{c} \mathbf{s}} (\overline{\rho q}), \tag{64}$$

where \mathbf{s} is the space coordinate along the stream direction. The partial derivative with respect to \mathbf{s} is defined by

$$\frac{\hat{c}}{\hat{c} \mathbf{s}} (\overline{\rho q}) = \frac{u}{q} \frac{\hat{c}}{\hat{c} x} (\overline{\rho q}) + \frac{v}{q} \frac{\hat{c}}{\hat{c} y} (\overline{\rho q}) + \frac{w}{q} \frac{\hat{c}}{\hat{c} z} (\overline{\rho q}), \tag{65}$$

The upwind density evaluation scheme given by Eqs. (64) and (65) is very similar to the artificial density scheme described in Section 3.3, see in particular Eq. (43a) and (43b). Following the work of Jameson [369] a second-order-accurate numerical evaluation of Eq. (64) can be written. For brevity it is given in one dimension as

$$(\hat{\rho q})_{i-1/2} \cong (\rho q)_{i-1/2} - \Delta(\overline{\rho q})_{i-1/2} + \Psi(r) \Delta(\overline{\rho q})_{i-1/2},$$

where Δ is a backward difference operator and $\Psi(r)$ is a limiter function defined by

$$\Psi(r) = \max[\min(r, 1), 0],$$

and r is the ratio of flux gradients at $i - 1/2$ and $i + 1/2$. At a flow extremum r will be negative causing $\Psi = 0$. Thus, a dissipative first-order scheme is produced. At all other flow points a second-order-accurate upwind scheme is realized.

The FEM discretization used by Kinney et al. [365] can be expressed by multiplying the upwind Cartesian form of the full potential equation by a test function N and then by integrating over an appropriate control volume Ω

$$- \iiint_{\Omega} [\hat{\rho}(\phi_x N_x + \phi_y N_y + \phi_z N_z)] d\Omega + \iint_S (\hat{\rho} \nabla \phi \cdot \mathbf{n}) N dS = 0,$$

where S is the boundary of Ω . The surface integral in the above equation is zero at all solid surfaces and is replaced with an appropriate condition in the far field. The velocity potential and test function N are assumed to vary linearly across each FEM cell. To complete this formulation upwind fluxes are needed for each tetrahedron cell center. This is accomplished by first storing upwind fluxes at each node, which are obtained from a piecewise average of the fluxes in the upwind cells.

A simple two-dimensional example is shown in Fig. 29. The upwind flux in cell D is computed using the centered fluxes in cells A and B (both upwind to cell D). The formula used for this computation is given by

$$(\hat{\rho q})_D = (\rho q)_D - [a(\overline{\rho q})_A + b(\overline{\rho q})_B],$$

where a and b must sum to one and are computed from the local velocity vector and geometric aspects of the

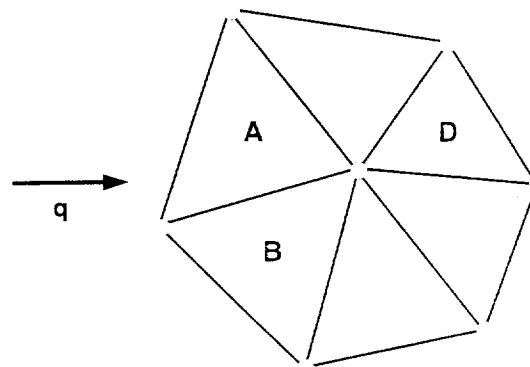


Fig. 29. Sketch of a typical two-dimensional upwind flux calculation for cell D in terms of the centered fluxes in the upwind cells A and B.

local cell. This is just a first-order-accurate example without flux limiters. For the complete flux computation algorithm see Kinney et al. [365].

Once the spatial discretization is complete an iteration scheme must be constructed that remains hyperbolic in supersonic regions of flow and elliptic in subsonic regions. Carefully evaluated upwind-biased temporal damping terms of the form ϕ_{α_i} must be added for stability in supersonic regions, just like other potential iteration schemes (see Jameson [81] for more discussion on this last point). A Newton linearization is performed about the previous iteration involving only the first-order terms with the second-order terms lagged one iteration. The result of the linearization is a large sparse linear system that must be solved during each iteration using an appropriate linear solver. A typical transonic wing computation using this approach on a grid consisting of 99,302 nodes and 542,624 tetrahedra requires about 22 min of CPU time and 12 MW of memory on a single processor of a Cray C-90 computer.

A significant advantage of the unstructured grid approach is the ability to perform computations on complex geometries relatively easily, i.e., with this approach the volume grid generation issues are simplified for complex shapes. An example taken from Kinney et al. [368] showing a grid and surface solution for a geometrically complete transport aircraft is displayed in Fig. 30. The aircraft configuration consists of a wing, body, pylon, nacelle and a vertical tail. The simulation conditions consist of a free-stream Mach number of 0.78 and a lift coefficient of 0.44. Note the fine grid resolution on the aircraft surface in Fig. 30a. In Fig. 30b, the surface contours for the pressure coefficient are displayed. Note the strong transonic shock on the aft-part of the wing upper surface.

3.11.3. Unstructured Cartesian grid methods

As already presented, unstructured Cartesian grid methods utilize grids composed of equal-sized squares in two dimensions or equal-sized cubes in three dimensions. Solution adaptivity is achieved by subdividing each grid cell in a region of high solution gradient into four smaller equal-sized squares for two dimensions or into eight smaller equal-sized cubes for three dimensions. Grid adaptation can also be performed in regions where the geometry is highly curved and thus likely to produce large solution gradients. This process of cell subdivision typically continues for several levels until an accuracy requirement is achieved or until a predetermined maximum number of subdivisions is attained. The subdivision process generates discontinuous cell sizes between adjacent cells, but never more than a two-to-one level of discontinuity. The flow solver discretization scheme, typically of the FEM variety, is implemented to handle the adjacent cell discontinuity. In addition, Cartesian grid cells that intersect an aerodynamic surface result in irregular cells that must also be handled within the framework of the FEM scheme. This approach for handling

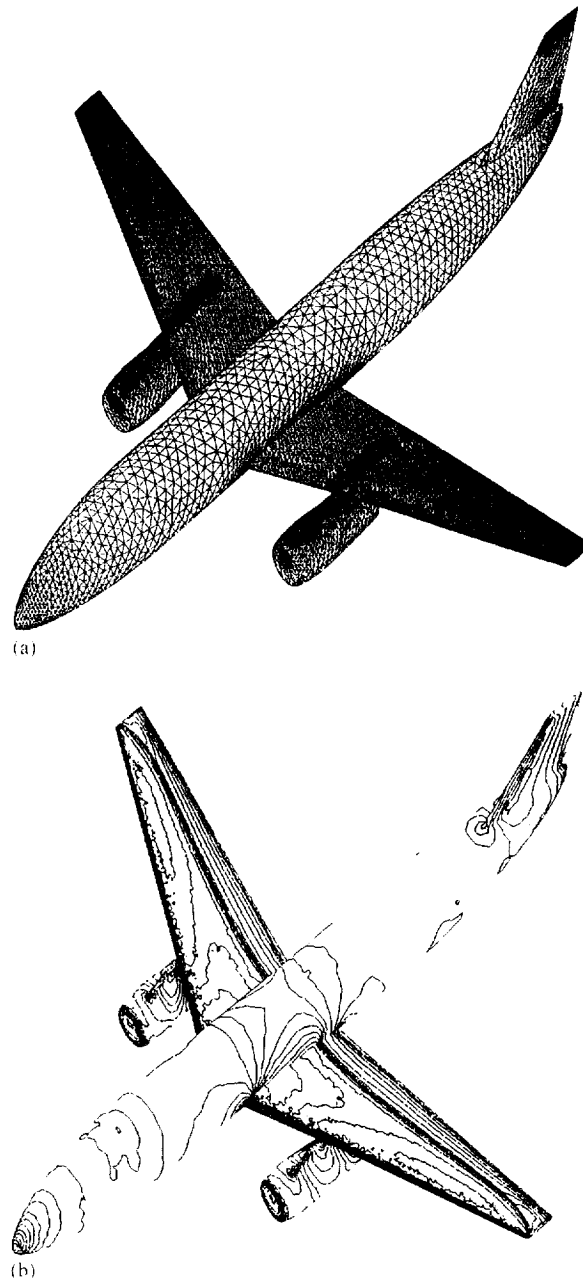


Fig. 30. Solution about a typical twin-engine transport aircraft in the transonic flow regime using the unstructured FEM approach. $M_\infty = 0.78$, $C_L = 0.44$, taken from Kinney et al. [368]. (a) Surface grid. (b) Surface pressure coefficient contours.

complex geometries is perhaps the most general and easiest to implement of all the approaches utilized in CFD. Its ease of implementation and generality are derived from the simple manner in which the intersection between an analytically defined Cartesian grid and an arbitrary CAD-defined geometry can be computed. The

biggest disadvantage of this approach occurs when a flow field gradient exists at a 45° angle with respect to the grid, e.g., the gradient along a swept-wing leading edge. In this case the grid must be refined in two (or sometimes even three) directions to resolve the gradient.

A number of unstructured Cartesian grid full potential applications have been presented in the literature for a variety of transonic flow problems. Early applications can be found in Johnston et al. [370,371] and Wedan and South [372] for simple geometries. In these studies the basic theoretical aspects including irregular cell treatment at boundaries are emphasized. Another unstructured Cartesian approach used for solving three-dimensional inlets is presented by Brown [189]. In this study the nonconservative full potential equation is solved using a finite-difference multigrid scheme. A recent Cartesian unstructured grid method for solving the full potential equation based on a Newton–Krylov–Schwarz scheme is studied in Cai et al. [373]. This type of method employs a domain decomposition FEM approach and thus is suitable for parallel computer implementation. Total solution time for transonic flow cases is reported to be six times larger than for subsonic cases. A recent approach utilizing the unstructured Cartesian grid procedure, that has been extensively developed and used for many applications, is the approach used in the TRANAIR code. The theoretical aspects of the TRANAIR method including the FEM discretization procedure are described in detail in Rubbert et al. [374], Young et al. [375] and Bieterman et al. [376]. Numerous complex geometry applications utilizing TRANAIR have been reported in the literature. A few of these include Cenko and Piranian [377] for store loads prediction on fighter aircraft, Ridlon et al. [378] for static aeroelastic analysis, SenGupta et al. [379] for unsteady aerodynamic and flutter computations, Madson [380] and Goodsell et al. [381] for fighter aircraft computations, Chen et al. [382] for engine-airframe integration applications with and without power, Madson [383] for supersonic flow sonic-boom computations and Jou et al. [384] for aerodynamic design optimization.

Computational costs associated with the TRANAIR program for typical transonic analysis computations increase approximately as $O(N^{1.2})$, where N is the number of elements used in the problem (Young et al. [375]). A typical transonic computation consisting of 200,000 elements requires about 3500 s of CPU time on a Cray X-MP computer for a tightly converged solution. This consists of about 1000 s of setup time (including grid generation) and about 2500 s for flow solution time. The memory requirements for TRANAIR increase approximately as $O(N)$. For a typical transonic computation consisting of 200,000 elements the memory requirement is just over 50 million words.

A typical result from the TRANAIR code taken from Jou et al. [384] is presented in Fig. 31. This example

shows TRANAIR's complex geometry handling capability in the context of a propulsion-airframe-integration design optimization application. Fig. 31a shows Mach number contours on the upper surface of a transonic wing in the vicinity of a low-mounted nacelle before design optimization. Note the existence of a transonic shock emanating from the strut-wing juncture. Fig. 31b shows the same configuration after design optimization modifications have been made. As can be seen the upper-wing-surface shock caused by a propulsion interference effect has been removed in Fig. 31b.

4. Concluding remarks

Numerical solution of nonlinear potential equations for transonic cruise analysis and design has received much attention within the CFD research community in the last 20–30 years and has reached a mature level of development in most application areas. This review describes the key historical milestones in this developmental process and provides a quantitative description of existing nonlinear potential equation simulation capabilities. Throughout the review computational results with experimental comparisons are presented to highlight key discussion points and to demonstrate method capabilities.

Specific summarizing comments from this review are presented as follows:

- (1) To begin, this review presents a detailed description of several nonlinear potential formulations with emphasis on the full potential equation. This includes a discussion of derivation assumptions, boundary conditions, transformation techniques and conservative versus nonconservative issues. These formulations are all isentropic and irrotational. Nevertheless, shock waves can be captured using a nonlinear potential numerical algorithm with good agreement to the more exact Euler equations providing the shock waves are weak, i.e., the shock-normal component of Mach number just upstream of a shock wave should not exceed about 1.3.
- (2) A variety of authors have demonstrated nonunique solutions for the conservative full potential equation for two-dimensional airfoil applications. The nonuniqueness manifests itself in the form of multiple lift values for a single angle of attack. The nonuniqueness only exists over a narrow range of freestream Mach number for transonic flow conditions. It has not been demonstrated for typical three-dimensional computations in any Mach number range, and thus, represents only an academic concern as the vast majority of nonlinear potential flow applications are three dimensional in nature.

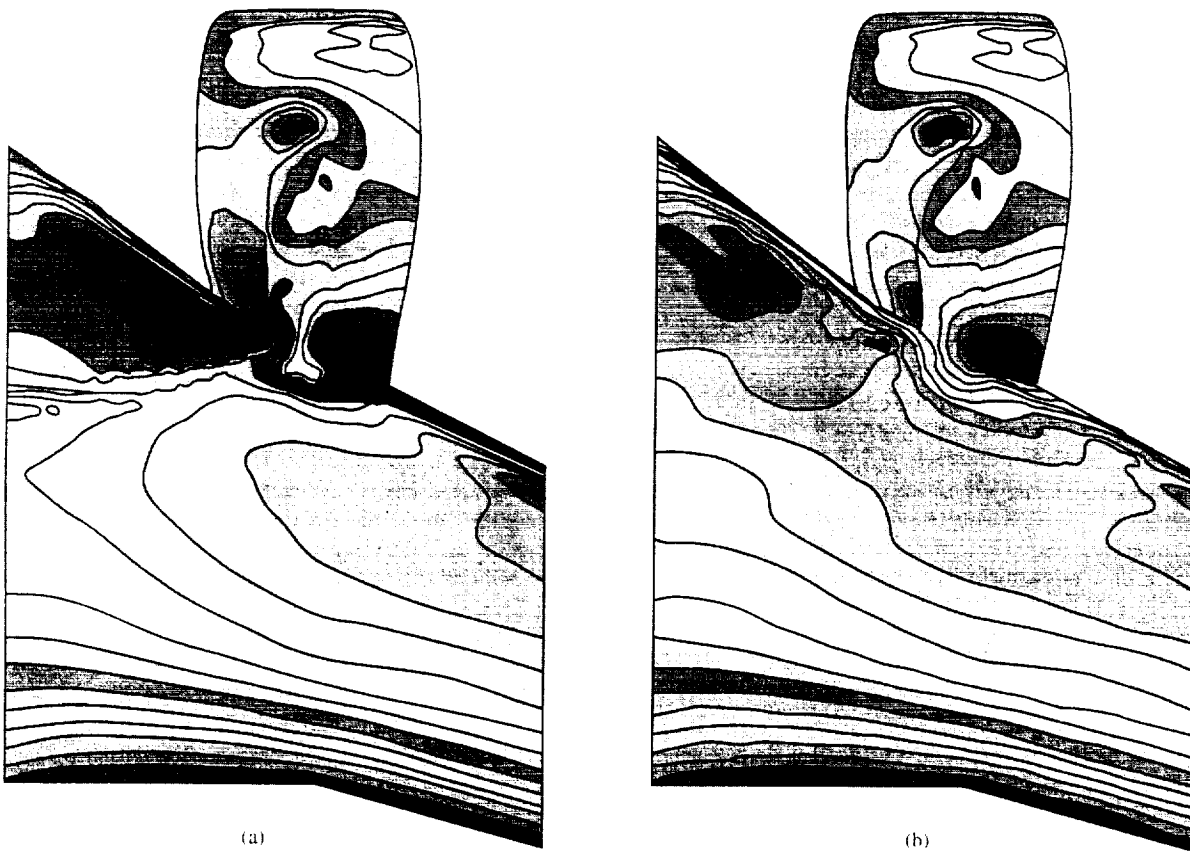


Fig. 31. Mach number contours on the wing upper surface in the vicinity of a nacelle showing numerical solutions before and after optimization, taken from Jou et al. [384]. (a) Before design optimization. (b) After design optimization.

- (3) A simple and easy-to-include nonisentropic potential flow correction procedure is available and has been demonstrated in a variety of steady and unsteady flow applications. One variation of this correction procedure, which requires an algorithm change only at shock waves, typically corrects the shock wave's strength and position producing good agreement with corresponding Euler solutions, even for shocks that violate the weak shock wave assumption. Other variations allow corrections for both entropy and vorticity. In one application the nonisentropic correction procedure is used to eliminate the nonuniqueness of the conservative full potential equation for airfoil computations.
- (4) The main current application area for nonlinear potential formulations is transonic cruise analysis for all types of aerospace vehicles, especially transport aircraft. Other important application areas include rotorcraft rotor analysis, including the effects of a modeled rotor wake; transonic cruise design and/or optimization especially the minimization of wing/fuselage and wing/fuselage pylon nacelle interference

effects; aeroelastic computations including the prediction of flutter boundaries for transonic wings; and analysis of forebody or slender wing/body configurations for low supersonic flows. Many of these applications include a direct viscous correction procedure utilizing either a momentum integral or a full boundary layer equation approach.

- (5) There are many different algorithms in use for solving nonlinear potential equations. Generally, an algorithm consists of a spatial discretization scheme, which determines spatial accuracy, and an iteration scheme, which determines steady-state convergence efficiency (for steady problems) or time accuracy (for unsteady problems). Typical spatial discretization schemes include an artificial viscosity, upwind flux or artificial density upwinding method cast in the framework of a finite-difference, finite-volume or finite-element approach. Iteration schemes include classical relaxation methods, e.g., SOR or SLOR, and more recently developed schemes, e.g., approximate factorization, multigrid, minimum residual or conjugate gradient methods. Space marching algorithms are

also used for problems with supersonic freestream flows. In this type of algorithm the cross-flow plane solution is obtained via a local iteration scheme that resembles a two-dimensional transonic flow relaxation algorithm. Then the three-dimensional solution is obtained without global iteration by marching downstream. The outer bow shock is obtained via a shock capturing or shock fitting scheme as the algorithm is marched downstream. In one application a hybrid approach is presented that utilizes a marching scheme in supersonic flow regions coupled with an embedded local relaxation scheme for subsonic pockets of flow.

- (6) Nonlinear potential methods are used extensively in design and optimization. With their relatively short turnaround time on high-end desktop computers, they are ideally suited for the repetitive parametric variations required for the design environment. Methods reviewed in this section include indirect, inverse, gradient optimization and genetic optimization methods. Inverse and gradient optimization methods are both well established and utilized heavily. Genetic optimization methods, which are quite expensive, but theoretically work in noisy design spaces with multiple local extrema, are just beginning to be explored.
- (7) A variety of nonlinear potential methods have been extensively developed for simulating the flow over geometrically complex configurations. These methods include various zonal-grid approaches (both patched and chimera), unstructured approaches and Cartesian unstructured approaches. The unstructured approaches are generally more accommodating in the treatment of complex configurations, but are less computationally efficient. Several complex geometry applications involving propulsion airframe integration with propulsive effects are demonstrated.
- (8) Three-dimensional nonlinear potential solver CPU times are difficult to quantify because there are many contributing factors that may cause large variations. A few of these include the grid density level, configuration complexity, numerical scheme variations (especially the iteration scheme) and the discretization approach. However, most three-dimensional nonlinear potential solver CPU times range from 1 min to 1 h on current high-end computer systems. The smaller run times typically involve coarse-grid structured approaches for simple configurations and the larger run times involve fine-grid unstructured approaches for more complex configurations. Although there are few quantitative head-to-head comparisons between different formulations, the cost for performing three-dimensional transonic flow simulations based on the full potential formulation is typically an order of magnitude less than for a comparable simulation using the Euler equations.

5. Recommendations for future work

Despite the advanced state of development for numerical solutions of the full potential equation, there are several areas that require improvements. First, geometric-handling computer software must be more flexible and automated to a higher degree, minimizing (or even eliminating) the amount of human intervention required to move a new CAD geometry into the CFD environment. Of course, this is not just a full potential issue, but an area of improvement required for all CFD formulations. Advances in this area are largely paced by developments in surface-geometry representation and surface grid generation. Nevertheless, enough research needs to be conducted in the full potential arena so that new geometric handling improvements are absorbed by the full potential research community.

Full potential analysis and design methodologies need to be better integrated into design environments, especially environments with a hierarchy of tools. The aerospace vehicle designer should be able to choose between more expensive Euler/Navier–Stokes formulations when the physics dictates and the schedule and budget allow or approximate but faster potential formulations when schedule and budget dictate and the physics allows. This process should be performed in a seamless fashion using universal surface and volume grid generators and universal post processing software.

Optimization methodologies should be researched. The genetic algorithm approach is extremely promising, but largely undeveloped for aerodynamic optimization problems. Obviously, this is because of its inherent expense and lack of knowledge for how to apply genetic algorithm theory to aerodynamic and/or multi-discipline design. The same situation existed for gradient-based optimization methods fifteen years ago and now with improvements in sensitivity derivative computation and a 100-fold increase in computer power this technique is widely accepted for aerodynamic design and optimization. As design spaces become more detailed and include more disciplines, the genetic optimization approach may provide an attractive alternative for optimization. Utilization of a fast full potential analysis capability in conjunction with genetic optimization may provide an attractive research approach to help develop genetic optimization methodologies.

Full potential analysis tools should be more completely integrated into the conceptual design environment. Because they are quantitative (for at least cruise conditions) and fast, potential methods could provide improved parametric aerodynamic results for many conceptual configurations very efficiently. For example, utilizing newer parallel computers a wing/fuselage parametric variation consisting of 1000 transonic flow cases might require only about 10–15 min of CPU time.

Unsteady applications utilizing the full potential formulation should be more completely developed, especially those applications that are untenable using more complete formulations. A good example of this is the helicopter blade-vortex-interaction application.

Hybrid applications involving the full potential formulation coupled with other formulations is an important area that needs additional research. For example, utilization of the fast full potential formulation in outer regions of flow to drive convergence and the Navier–Stokes formulation near all surfaces to capture viscous effects could produce accurate separated-flow physics at a fraction of the cost.

The nonuniqueness problem associated with numerical solutions of the conservative nonlinear potential formulations for two-dimensional applications, which appears to be formulational and not numerical in nature, needs to be explained.

Acknowledgements

The author extends a special thanks to Drs. Dennis Jespersen, Frank Caradonna, Samson Cheung and David Kinney for the considerable time they spent in reviewing this manuscript. Their many helpful comments and suggestions have greatly added to this survey and are greatly appreciated. The author also thanks Dr. Argyris Panaras for suggesting this review article.

References

- [1] Ballhaus Jr WF, Steger JL. Implicit approximate factorization schemes for the low-frequency transonic equation. NASA TM X-73082, 1975.
- [2] Stone HL. Iterative solution of implicit approximations of multi-dimensional partial differential equations. *SIAM J Num Anal* 1968;5:530–58.
- [3] Flores J, Barton J, Holst T, Pulliam T. Comparison of the full-potential and Euler formulations for computing transonic airfoil flows. NASA TM 85983, 1984.
- [4] Rubbert PE. CFD and the changing World of airplane design. AIAA Wright Brothers Lecture, Anaheim, CA, 1994.
- [5] Hall MG. Computational fluid dynamics — a revolutionary force in aerodynamics. AIAA Paper 81-1014, 1981.
- [6] South Jr JC. A historical perspective and overview of computational aerodynamics. In: Habashi W, editor. *Advances in Computer Transactions*. Swansea, UK: Pineridge Press, Ltd., 1985. p. 1–20.
- [7] Holst TL, Slooff JW, Yoshihara H, Ballhaus Jr WF. Applied computational transonic aerodynamics. AGARDograph No. 266, 1982.
- [8] Kordulla W. Calculation of 3-D transonic flows. VKI Lecture Series in CFD. 1980:1980–6.
- [9] Nixon D, Kerlick G. Potential equation methods for transonic flow prediction. In: Nixon D, editor. *Transonic Aerodynamics*. Progress in Astronautics and Aeronautics, Vol. 81. New York: AIAA, 1982.
- [10] Nixon D, editor. *Transonic aerodynamics*. Prog. in Astro. and Aero. Vol. 81. New York: AIAA, 1982.
- [11] Caughey DA, Hafez MM, editors. *Frontiers of computational fluid dynamics*. West Sussex, England: Wiley, 1994.
- [12] Zierep J, Oertel H, editors. *Symposium transsonicum III*. New York: Springer, 1988.
- [13] Habashi W, editor. *Advances in computational transonics*. Swansea, UK: Pineridge Press Ltd., 1985.
- [14] Henne PA, editor. *Applied computational aerodynamics*. Progress in Astronautics and Aeronautics, Vol. 125. Washington, DC: AIAA, 1990.
- [15] Hirsch C. *Numerical computation of internal and external flows, vol. 2. Computational methods for inviscid and viscous flows*. West Sussex, England: Wiley, 1990.
- [16] Anderson DA, Tannehill JC, Pletcher RH. *Computational fluid mechanics and heat transfer*. Washington, DC: Hemisphere, 1984.
- [17] Pai SI, Luo S. *Theoretical and computational dynamics of a compressible flow*. New York: Van Nostrand Reinhold, 1991.
- [18] Owczarek J. *Fundamentals of Gas Dynamics*. Scranton, PA: Inter. Textbook Company, 1964.
- [19] Smith AMO. The panel method: its original development. In: Henne P, editor. *Applied computational aerodynamics*. Progress in astronautics and aeronautics, vol. 125. Washington, DC: AIAA, 1990. p. 3–21.
- [20] Hess JL. Linear potential schemes, Applied computational aerodynamics. In: Henne P, editor. *Progress in Astronautics and Aeronautics*, Vol. 125. Washington, DC: AIAA, 1990. p. 21–38.
- [21] Hoeijmakers H. Panel methods for aerodynamic analysis and design, AGARD-R-783, Paper 5, 1992.
- [22] Roggero F, Larguier R. Aerodynamic calculation of complex three-dimensional configurations. *J Aircr* 1993;30:561–70.
- [23] Ballhaus Jr WF, Goorjian PM. Implicit finite-difference computations of unsteady transonic flows about airfoils. *AIAA J* 1977;15:1728–35.
- [24] van der Vooren J, Slooff JW, Huizing GH, Van Essen A. Remarks on the suitability of various transonic small perturbation equations to describe three-dimensional transonic flow-examples of computations using a fully conservative rotated difference scheme. *Symposium Transonicum II*. Berlin: Springer, 1975.
- [25] Slooff J. General theory. *Appl Comp Tran Aero*, AGARD-AG-266, 1982. p. 2–18.
- [26] Ames WF. *Numerical methods for partial differential equations*. New York: Academic Press, 1977.
- [27] Mitchell AR. *Computational methods for partial differential equations*. New York: Wiley, 1969.
- [28] Lapidus A. A detached shock calculation by second-order finite-differences. *J Comp Phys* 1967;2:154–77.
- [29] Viviani H. Formes Conservatives des Equations de la Dynamique des Gaz. *La Recherche Aerospaciale* 1974;1: 65–8.
- [30] Vinokur M. Conservative equations of gas dynamics in curvilinear coordinate systems. *J Comp Phys* 1974;14:105–25.

- [31] Steger JL. Implicit finite difference simulations of flow about arbitrary geometries with applications to airfoils. AIAA Paper 77-665, 1977.
- [32] NACA. Equations, tables, and charts for compressible flow. NACA Report 1135, 1953.
- [33] Steger JL, Baldwin BS. Shock waves and drag in the numerical calculation of isentropic transonic flow. NASA TN D-6997, 1972.
- [34] Lax P. Weak solutions of non-linear hyperbolic equations and their numerical calculation. *Comm Pure and Appl Math* 1954;7:159–93.
- [35] Gregg RD, Henne PA. An assessment of mass conservation for potential flow shock simulation. AIAA Paper 82-158, 1982.
- [36] Newman PA, South Jr JC. Conservative versus nonconservative differencing: transonic streamline shape effects. NASA TM X-72827, 1976.
- [37] Newman PA, South Jr JC. Influence of nonconservative differencing on transonic streamline shapes. *AIAA J* 1976;14:1148–9.
- [38] Steinhoff J, Jameson A. Multiple solutions of the transonic potential flow equation. AIAA Paper 81-1019, 1981.
- [39] Salas MD, Gumbert CR, Turkel E. Nonunique solutions to the transonic potential flow equation. *AIAA J* 1984;22:145–6.
- [40] Salas MD, Jameson A, Melnik RE. A comparative study of the nonuniqueness problem of the potential equation. AIAA Paper 83-1888, 1983.
- [41] Salas MD, Jameson A, Melnik RE. A comparative study of the nonuniqueness problem of the potential equation. NASA TP-2385, 1985.
- [42] Dowell EH, Bland SR, Williams MH. Linear/nonlinear behavior of unsteady transonic aerodynamics. *AIAA J* 1983;21:38–46.
- [43] Williams MH, Bland SR, Edwards JW. Flow instabilities in transonic small-disturbance theory. *AIAA J* 1985;23:1491–6.
- [44] Murty HS. Nonunique solutions in unsteady transonic flow. *J Aircr* 1993;30:129–31.
- [45] Holst TL. Unpublished results, 1998.
- [46] Zi-qiang Z, Xue-Song B. Numerical computation of improved transonic potential method. *J Aircr* 1992;29:180–4.
- [47] McGrattan K. Comparison of transonic flow models. *AIAA J* 1992;30:2340–3.
- [48] Murman E, Cole J. Calculation of plane steady transonic flows. *AIAA J* 1971;9:114–21.
- [49] Bailey FR. Numerical calculation of transonic flow about slender bodies of revolution. NASA TN D-6582, 1971.
- [50] Krupp JA, Murman EM. The numerical calculation of steady transonic flows past thin lifting airfoils and slender bodies. *AIAA J* 1972;10:880–6.
- [51] Bailey FR, Steger JL. Relaxation techniques for three-dimension transonic flow about wings. AIAA Paper, 1972:72-189.
- [52] Ballhaus Jr WF, Bailey FR. Numerical calculations of transonic flow about swept wings. AIAA Paper, 1972:72-677.
- [53] Newman PA, Klunker FB. Computation of transonic flow about finite lifting wings. *AIAA J* 1972;10:971–3.
- [54] Murman EM. Analysis of embedded shock waves calculated by relaxation methods. *AIAA J* 1974;12:626–33.
- [55] Bailey FR, Ballhaus Jr WF. Comparison of computed an experimental pressures for transonic flows about isolated wings and wing-fuselage configurations. NASA SP-347, 1975.
- [56] Rolfs S, Vanino R. A steady relaxation method for two- and three-dimensional transonic flows. *Euromech 40, Transonic Aerodynamics, Saltsjobaden, Sweden*, 1973.
- [57] Schmidt W, Hedman S. Recent explorations in relaxation methods for three-dimensional transonic potential flow. *ICAS Paper 76-22*, 1976.
- [58] van der Vooren J, Huizing GH, Van Essen A. A finite difference method for the calculation of transonic flow about a wing, based on small perturbation theory. NLR TR 81031 L, 1981.
- [59] Mason W, Mackenzie DA, Stern MA, Johnson JK. A numerical three-dimensional viscous transonic wing-body analysis and design tool. AIAA Paper 78-101, 1978.
- [60] Albone CM, Hall MG, Joyce G. Numerical solutions for transonic flows past wing-body configurations. *Symposium Transonicum II*. Berlin: Springer, 1976.
- [61] Firman MCP. Calculations of transonic flow over wing/body combinations with an allowance for viscous effects. AGARD CP-291, Paper 8, 1981.
- [62] Rae WJ. Calculation of three-dimensional transonic compressor flow fields by a relaxation method. *J Energy* 1977;1:284–96.
- [63] Rae WJ, Lordi JA. A study of inlet conditions for three-dimensional transonic compressor flows. *Calspan Report No. XE-6129-A-4*, 1978.
- [64] Shankar V, Malmuth ND. Computational treatment of three-dimensional canard-wing interactions. *J Aircr* 1983;20:456–61.
- [65] Phillips PS, Waggoner EG. Transonic wind-tunnel wall interference prediction code. *J Aircr* 1990;27:915–6.
- [66] Boppe C. Calculation of transonic wing flows by grid embedding. AIAA Paper 77-207, 1977.
- [67] Boppe CW, Stern MA. Simulated transonic flows for aircraft with nacelles, pylons, and winglets. AIAA Paper 80-130, 1980.
- [68] Garabedian PR. Estimation of the relaxation factor for small mesh size. *Math Tab Natl Res Coun* 1956;10:183–5.
- [69] Steger JL, Lomax H. Numerical calculation of transonic flow about two-dimensional airfoils by relaxation procedures. *AIAA J* 1972;10:49–54.
- [70] Garabedian P, Korn D. Analysis of transonic airfoils. *Comm Pure and Appl Math* 1971;24:841–51.
- [71] Bauer F, Garabedian P, Korn D, Jameson A. *Supercritical Wing Section II*. Lecture Notes in Economics and Mathematical Systems 108. New York: Springer, 1975.
- [72] Carlson LA. Transonic airfoil analysis and design using Cartesian coordinates. *J Aircr* 1976;13:359–66.
- [73] South Jr JC, Jameson A. Relaxation solutions for inviscid axisymmetric transonic flow over blunt or pointed bodies. *First AIAA CFD Conference*, Palm Springs, CA, 1973.
- [74] Keller JD, South Jr JC. RAXBOD: A fortran program for inviscid transonic flow over axisymmetric bodies. NASA TM X-72831, 1976.
- [75] Ives DC, Liutermoza JF. Analysis of transonic cascade flow using conformal mapping and relaxation techniques. *AIAA J* 1977;15:647–52.

- [76] Ives DC, Liutermoza JF. Second-order-accurate calculation of transonic flow over turbomachinery cascades. *AIAA J* 1979;17:870–6.
- [77] Arlinger B. Calculation of transonic flow around axisymmetric inlets. *AIAA J* 1975;13:1614–21.
- [78] Baker TJ. A numerical method to compute inviscid transonic flow around axisymmetric ducted bodies. Symposium Transonicum II. Berlin: Springer, 1975.
- [79] Caughey DA, Jameson A. Accelerated iterative calculation of transonic nacelle flow fields. *AIAA J* 1977;15:1474–80.
- [80] Reyhner T. Cartesian mesh solution for axisymmetric transonic potential flow around inlets. *AIAA J* 1977;15:624–31.
- [81] Jameson A. Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1. *Comm Pure Appl Math* 1974;27:283–309.
- [82] Jameson A, Caughey DA, Newman PA, Davis RM. A brief description of the Jameson–Caughey NYU transonic swept-wing computer program FLO-22. NASA TM X-73996, 1976.
- [83] Newman PA, Carter JE, Davis RM. Interaction of a two-dimensional strip boundary layer with a three-dimensional transonic swept-wing code. NASA TM-78640, 1978.
- [84] Henne PA, Hicks RM. Wing analysis using a transonic potential flow computational method. NASA TM-78464, 1978.
- [85] Reyhner TA. Transonic potential flow around axisymmetric inlets and bodies at angle of attack. *AIAA J* 1977;15:1299–306.
- [86] Forcey CR, Carr MP. The calculation of transonic flow over three-dimensional swept wings using the exact potential equation. ARA Memo 207, 1978.
- [87] Chang I, Tauber M. Numerical calculation of the transonic potential flow past a cranked wing. NASA TM 84391, 1983.
- [88] Chang I. Transonic flow analysis for rotors—I. Three-dimensional, quasi-steady full potential calculation. NASA TP-2375, 1984.
- [89] Chang I, Tung C. Numerical solution of the full potential equation for rotors and oblique wings using a new wake model. AIAA Paper 85-0268, 1985.
- [90] Arieli R, Tauber ME, Saunders DA, Caughey DA. Computation of transonic flow about helicopter rotor blades. *AIAA J* 1986;24:722–7.
- [91] Egolf TA, Sparks SP. Hovering rotor airload prediction using a full potential flow analysis with realistic wake geometry. 41st Annual Forum of the AHS, Ft. Worth TX: 1985.
- [92] Jameson A. Transonic potential flow calculation using conservative form. Second AIAA CFD Conference, 1975, p. 148–155.
- [93] Jameson A, Caughey DA. A finite-volume method for transonic potential flow calculations. Third AIAA CFD Conference, 1977, p. 35–54.
- [94] Caughey DA, Jameson A. Numerical calculation of transonic potential flow about wing-body combinations. *AIAA J* 1979;17:175–81.
- [95] Caughey DA, Jameson A. Progress in finite-volume calculations for wing-fuselage combinations. *AIAA J* 1980;18:1281–8.
- [96] Verhoff A, O'Neil PJ. Extension of FLO codes to transonic flow prediction for fighter configurations. MCAIR Report 81-004, 1981.
- [97] Streett C. Viscous-inviscid interaction for transonic wing-body configurations including wake effects. *AIAA J* 1982;20:915–23.
- [98] Woodson SH, Campbell JF, DeJarnette FR. Interactive three-dimensional boundary-layer method for transonic flow over swept wings. *AIAA J* 1991;29:678–9.
- [99] Thompson JF, Thames FC, Mastin CW. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *J Comp Phys* 1974;15:288–319.
- [100] Yu NJ. Transonic flow simulations for complex configurations with surface fitted grids. AIAA Paper 81-1258, 1981.
- [101] Chen LT, Yu KC, Dang TQ. Transonic computational method for an aft-mounted nacelle/pylon with power effect. *J Aircr* 1990;27:878–85.
- [102] Hinson B, Burdges K. Acquisition and application of transonic wing and far-field test data for three-dimensional computational method evaluation. AFOSR TR-80-0421, 0422, 1980.
- [103] Chattot JJ, Coulombeix C, da Silva Tome C. Calculs d'Écoulements Transsoniques Autour d'Ailes. *La Recherche Aérospatiale* 1978;4:143–59.
- [104] Eberle A. A finite volume method for calculating transonic potential flow around wings from the pressure minimum integral. NASA TM-75324, 1978.
- [105] Chen LT, Caughey DA. Calculation of transonic inlet flow fields using generalized coordinates. *J Aircr* 1980;17:167–74.
- [106] Holst TL, Ballhaus Jr WF. Fast conservative schemes for the full potential equation applied to transonic flows. *AIAA J* 1979;17:145–52.
- [107] Hafez MM, Murman EM, South Jr JC. Artificial compressibility methods for numerical solution of transonic full potential equation. *AIAA J* 1979;17:838–44.
- [108] Wong YS, Hafez MM. Some iterative schemes for transonic potential flows. *AIAA J* 1985;23:808–10.
- [109] Farrell C, Adamczyk J. Full potential solution of transonic quasi-3-D flow through a cascade using artificial compressibility. ASME Paper 81-GT-70, 1981.
- [110] Akay HU, Ecer A. Transonic flow computations in cascades using finite element method. *ASME J. Eng. Power* 1981;103:657–64.
- [111] Deconinck H, Hirsch C. Finite element methods for transonic blade-to-blade calculation in turbomachines. *ASME J Eng Power* 1981;103:665–77.
- [112] Shankar V. Conservative full potential, implicit marching scheme for supersonic flows. *AIAA J* 1982;20: 1508–14.
- [113] Green LL, South Jr JC. Conservative full-potential calculations for axisymmetric, transonic flow. *AIAA J* 1983;21:1492–9.
- [114] Eberle A. Evaluation of a minimum principle for transonic flow computations by finite elements. GAMM Conference on Numerical Methods in Fluid Dynamics, Cölogne, West Germany, 1979.
- [115] Eberle A. Finite element methods for the solution of full potential equation in transonic steady and unsteady flow. Third International Conference on Finite Elements in Flow Probs. Banff, Alberta, Canada, 1980.

- [116] Steger JL, Caradonna FX. A conservative implicit finite difference algorithm for the unsteady transonic full potential equation. AIAA Paper 80-1368, 1980.
- [117] Goorjian PM. Implicit computations of unsteady transonic flow governed by the full potential equation in conservative form. AIAA Paper 80-0150, 1980.
- [118] Ecer A, Spyropoulos T. Block-structured solution scheme for analyzing three-dimensional transonic potential flows. AIAA J 1987;25:1292–300.
- [119] Neel RE. Advances in computational fluid dynamics: Turbulent separated flows and transonic potential flows. Ph.D. Thesis, Virginia Poly. Inst. and State University, Blacksburg, VA, 1997.
- [120] Holst T, Thomas S. Numerical solution of transonic wing flow fields. AIAA J 1983;21:863–70.
- [121] Holst TL. Conservative algorithm for solving the transonic full potential equation. AIAA J 1980;18:1431–9.
- [122] Holst TL. Multizone chimera algorithm for solving the full-potential equation. J Aircr 1998;35:412–21.
- [123] Engquist B, Osher S. Stable and entropy satisfying approximations for transonic flow calculations. Math Comp 1980;34:45–75.
- [124] Goorjian PM, Van Buskirk R. Implicit calculations of transonic flows using monotone methods. AIAA Paper 81-0331, 1981.
- [125] Goorjian PM, Meagher MC, Van Buskirk R. Monotone implicit algorithms for the small-disturbance and full potential equations applied to transonic flows. AIAA Paper 83-0371, 1983.
- [126] Boerstoeel JW. A multigrid algorithm for steady transonic potential flows around aerofoils using newton iteration. NASA CP 2202, 1981. p. 152–172.
- [127] Slooff J. Some new developments in exact integral equation formulations for sub- or transonic compressible potential flow. Inter. School of Appl. Aero., Amalfi, Italy, 1982.
- [128] Osher S, Hafez M, Whitlow Jr W. Entropy condition satisfying approximations for the full potential equations of transonic flow. Math Comp 1985;44:1–30.
- [129] Osher S, Whitlow Jr W, Hafez M. Entropy condition satisfying approximations for the full potential equations of transonic flow. NASA TM 85751, 1984.
- [130] Hafez MM, Whitlow Jr W, Osher S. Improved finite-difference schemes for transonic potential flow calculations. AIAA J 1987;25:1456–62.
- [131] Van Leer B. On the relation between the upwind-differencing schemes of Godunov, Engquist–Osher and Roe. ICASE Report 81-11, 1981.
- [132] Habashi WG, Hafez MM. Finite element solutions of transonic flow problems. AIAA J 1982;20:1368–76.
- [133] Volpe G, Jameson A. Transonic potential flow calculations by two artificial density methods. AIAA J 1988;26:425–9.
- [134] Dulikravich GS. Analysis of artificial dissipation models for the transonic full-potential equation. AIAA J 1988;26:1238–45.
- [135] Hafez MM, Lovell D. Transonic small disturbance calculations including entropy corrections, numerical and physical aspects of aerodynamic flows conference. Long Beach, CA, 1981.
- [136] Hafez MM, Lovell D. Entropy and vorticity corrections for transonic flows. AIAA Paper 83-1926, 1983.
- [137] Klopfer GH, Nixon D. Non-isentropic potential formulation for transonic flows. AIAA J 1984;22:770–6.
- [138] Bridgeman JO, Strawn RC, Caradonna FX. An entropy and viscosity corrected potential method for rotor performance prediction. 44th Annual Forum of AHS, Washington, DC, 1988.
- [139] Kinney DJ, Hafez MM. Finite element computations of transonic potential flow. International Conference on Finite Elements in Fluids, Venezia, Italy, 1995.
- [140] Batina JT. Unsteady transonic small-disturbance theory including entropy and vorticity effects. J Aircr 1989;26:531–8.
- [141] Siclari MJ, Visich M. Shock fitting in conical supersonic full potential flows with entropy effects. AIAA Paper 84-0261, 1984.
- [142] Siclari MJ, Rubel A. Entropy corrections to supersonic conical nonlinear potential flows. AIAA Paper 84-1683, 1984.
- [143] Fuglsang DF, Williams MH. Non-isentropic unsteady small disturbance theory. AIAA Paper 85-0600, 1985.
- [144] Whitlow W. Application of a nonisentropic full potential method to AGARD standard airfoils. AIAA Paper 88-0710, 1988.
- [145] Bridgeman JO, Strawn RC, Caradonna FX, Chen CS. Advanced rotor computations with a corrected potential method. 45th Annual Forum of AHS, Boston, MA, 1989.
- [146] Chen CS, Bridgeman JO. Three-dimensional viscous rotor flow calculations using a viscous-inviscid interaction approach. NASA TM 102235, 1990.
- [147] Pulliam TH, Steger JL. Implicit finite-difference simulation of three-dimensional compressible flow. AIAA J 1980;18:159–67.
- [148] Thomas P, Lombard C. The geometric conservation law — A link between finite-difference and finite-volume methods of flow computation on moving grids. AIAA Paper 78-1208, 1978.
- [149] Hindman RC. Geometrically induced errors and their relationship to the form of the governing equations and the treatment of generalized mappings. AIAA J 1982;20:1359–67.
- [150] Flores J, Holst TL, Kwak D, Batiste DM. A new consistent spatial differencing scheme for the transonic full potential equation. AIAA J 1984;22:1027–34.
- [151] Thomas SD, Holst TL. A consistent spatial differencing scheme for the transonic full potential equation in three dimensions. NASA TM-86716, 1985.
- [152] Jiang H, Cai R. An accurate spatial differencing scheme for a three-dimensional full potential equation. AIAA J 1986;24:1057–8.
- [153] Peaceman DW, Rachford Jr HH. The numerical solution of parabolic and elliptic differential equations. J. SIAM 1955;3:28–41.
- [154] Douglas J. On the numerical integration of $U_{xx} + U_{yy} = U_1$ by implicit methods. J Soc Ind Appl Math 1955;3:42–65.
- [155] Douglas J, Gunn JE. A general formulation of alternating direction methods, Part I. Parabolic and Hyperbolic Problems. Numer Math 1964;6:428–53.

- [156] Yanenko NN. The method of fractional steps. Berlin: Springer, 1971.
- [157] Sankar NL, Tassa Y. An algorithm for unsteady transonic potential flow past airfoils, Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford Univ. and NASA Ames Research Center, 1980.
- [158] Ballhaus Jr WF, Jameson A, Albert J. Implicit approximate factorization schemes for the efficient solution of steady transonic flow problems. *AIAA J* 1978;16:573–9.
- [159] Holst TL. Implicit algorithm for the conservative transonic full potential equation using an arbitrary mesh. *AIAA J* 1979;17:1038–45.
- [160] Chattot JJ, Coulombeix C. Finite difference and finite element solutions of the full potential equation in two dimensions. Fourth GAMM Conference on Num Meth in Fluid Mech. Paris, France, 1981.
- [161] Roach RL, Sankar NL. The strongly implicit procedure applied to the flow field of transonic turbine cascades. *AIAA Paper* 81-0211, 1981.
- [162] Baker TJ. Potential flow calculation by the approximate factorization method. *J Comp Phys* 1981;42:1–19.
- [163] Chattot JJ. Calculation of three-dimensional unsteady transonic flows past helicopter blades. *NASA TP-1721*, 1980.
- [164] Baker TJ, Forsey CR. A fast algorithm for the calculation of transonic flow over wing/body combinations. *AIAA J* 1983;21:60–7.
- [165] Sankar NL, Malone JB, Tassa Y. An implicit conservative algorithm for steady and unsteady three-dimensional transonic potential flows. *AIAA Paper* 81-1016, 1981.
- [166] Catherall D. Optimum approximate-factorization schemes for two-dimensional steady potential flows. *AIAA J* 1982;20:1057–63.
- [167] South Jr JC, Hafez MM. Stability analysis of intermediate boundary conditions in approximate factorization schemes. *AIAA Paper* 83-1898, 1983.
- [168] Vadyak J, Atta EH. Approximate factorization algorithm for three-dimensional transonic nacelle/inlet flowfield computations. *J Prop Power* 1985;1:58–64.
- [169] Jialin Z. 3D transonic potential flow computation in an axial-flow compressor rotor by an approximate factorization scheme. *ASME Paper* 85-GT-16, 1985.
- [170] Cosentino GB, Holst TL. Numerical optimization design of advanced transonic wing configurations. *J Aircr* 1986;23:192–9.
- [171] Cheung S, Holst T. Aerodynamic shape optimization using a combined distributed/shared memory paradigm. Computational Aerosciences Workshop, Moffett Field, CA, 1998.
- [172] de Mattos BS. Numerischer Entwurf von transsonischen Flügeln und Tragflügelprofilen unter Anwendung des AF2-Algorithmus zur Lösung der vollständigen Potentialgleichung. Ph.D. thesis, Institut für Aerodynamik und Gasdynamik der Universität Stuttgart, 1995.
- [173] de Mattos BS, Wagner S. Numerical design of transonic wings in curvilinear coordinates. *AIAA Paper* 94-1897-CP, 1994.
- [174] Fedorenko RP. The speed of convergence on one iterative process. *USSR Comp Math and Math Phys* 1964;4:227–35.
- [175] Bakhvalov NS. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comp Math and Math Phys* 1966;6:101–35.
- [176] Nicolaides RA. On the L2 convergence of an algorithm for solving finite element systems. *Math Comp* 1977;31:892–906.
- [177] Hackbusch W. Convergence of multi-grid iterations applied to difference equations. Report 79-5, Köln University Mathematics Institute, Köln, West Germany, 1979.
- [178] Brandt A. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. *Lecture Notes in Physics* 18. Berlin: Springer, 1973. p. 82–89.
- [179] Brandt A. Multi-level adaptive solutions to boundary value problems. *Math Comp* 1977;31:333–90.
- [180] South Jr JC, Brandt A. Application of multi-level grid method to transonic flow calculations. *ICASE Report* 76-8, 1976.
- [181] Arlinger BG. Multi-grid technique applied to lifting transonic flow using full potential equation, *SAAB Report L-0-1 B 439*, 1978.
- [182] Jameson A. Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method. *AIAA Paper* 79-1458, 1979.
- [183] Fuchs L. Finite difference methods for plane steady inviscid transonic flows. *TRITA-GAD-2*, 1977.
- [184] Deconinck H, Hirsch C. A multigrid method for the transonic full potential equation discretized with finite elements on an arbitrary body fitted mesh. *NASA CP-2202*, 61–81, 1981.
- [185] Sankar NL. A multigrid strongly implicit procedure for transonic potential flow problems. *AIAA J* 1983;21:1481–2.
- [186] Arlinger BG. Axisymmetric transonic flow computations using a multigrid method. Seventh International Conference on Numeric Methods in Fluid Dynamics, Stanford University and NASA Ames Research Center, CA, 1980.
- [187] Volpe G. Multigrid approximate factorization scheme for two-element airfoil flows. *AIAA J* 1987;25:784–91.
- [188] McCarthy DR, Reyhner TA. A multi-grid code for three-dimensional transonic potential flow about axisymmetric inlets at angle of attack. *AIAA J* 1982;20:45–50.
- [189] Brown JJ. An embedded-mesh potential flow analysis. *AIAA J* 1984;22:174–8.
- [190] Shmilovich A, Caughey DA. Application of the multi-grid method to calculation of transonic potential flow about wing/fuselage combinations. *NASA CP-2202*, 1981. p. 101–130.
- [191] Caughey DA. Multi-grid calculation of three-dimensional transonic potential flows. *AIAA Paper* 83-374, 1983.
- [192] Chen LT, Vassberg JC, Peavey CC. A transonic wing-body flowfield calculation with improved grid topology. *AIAA J* 1985;23:1877–84.
- [193] van der Vooren J, van der Wees J, Meelker JH. MATRICS, transonic potential flow calculations about transport aircraft. *AGARD CP-412*, 1986.
- [194] van der Vooren J, van der Wees J. Inviscid drag prediction for transonic transport wings using a full-potential method. *J Aircr* 1991;28:869–75.
- [195] Shmilovich A, Caughey DA. Calculation of transonic potential flow past wing-tail-fuselage configurations using a multigrid technique. *J Aircr* 1985;22:581–6

- [196] Shmilovich A. Calculation of transonic flows with separation past arbitrary inlets at incidence. *J Aircr* 1990;28:601–9.
- [197] Cebeci T, Chen L, Chang K. An interactive scheme for three-dimensional transonic flows. In: Cebeci T, editor. *Num and Phys Aspects of Aero Flows III*. New York: Springer, 1986.
- [198] Sankar NL, Malone JB, Tassa Y. A strongly implicit procedure for steady three-dimensional transonic potential flow. *AIAA J* 1982;20:598–605.
- [199] Gordon F, Arieli R. Speed-up of the strongly implicit procedure with application to subsonic/transonic potential flows. *J Aircr* 1990;28:1345–6.
- [200] Malone JB, Sankar NL. Numerical simulation of two-dimensional unsteady transonic flows using the full potential equation. *AIAA J* 1984;22:1035–41.
- [201] Sankar NL, Malone JB. Solution techniques for the steady and unsteady transonic full potential equations. In: Habashi W, editor. *Adv Comp Trans*. Swansea, UK: Pineridge Press Ltd., 1985. p. 257–87.
- [202] South Jr JC, Keller JD, Hafez MM. Vector processor algorithms for transonic flow calculations. *AIAA J* 1980;18:786–92.
- [203] Bristeau M, Glowinski R, Periaux J, Perrier P, Pironneau O. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods: I least squares formulation and conjugate gradient solution of the continuous problem. *Comm Meth Appl Mech Eng* 1979;17(18):619–57.
- [204] Glowinski R, Periaux J, Pironneau O. Transonic flow simulation by the finite element method via optimal control. *Third Internernational Symposium Finite Elements in Flow Probs.*, Canada, 1980.
- [205] Wong YS, Hafez MM. Application of conjugate gradient methods to transonic finite difference and finite element calculations. *AIAA J* 1982;20:1526–33.
- [206] Wong YS, Hafez MM. A minimum residual method for transonic potential flow. *ICASE Report* 82-15, 1982.
- [207] Wong YS. Newton-like minimal residual methods applied to transonic flow calculations. *AIAA J* 1985;23:515–21.
- [208] Grossman B. Numerical procedure for the computation of irrotational conical flows. *AIAA J* 1979;17:828–37.
- [209] Grossman B, Siclari MJ. Nonlinear supersonic potential flow over delta wings. *AIAA J* 1981;19:573–81.
- [210] Siclari MJ. Supersonic nonlinear potential flow with implicit isentropic shock fitting. *AIAA J* 1982;20:924–32.
- [211] Bradley PF, Dwoyer DL, South Jr JC, Keen JM. Vectorized schemes for conical potential flow using the artificial density method. *AIAA J* 1986;24:13–20.
- [212] Sritharan SS, Seebass AR. Finite area method for nonlinear supersonic conical flows. *AIAA J* 1984;22:226–33.
- [213] Walkley KB, Smith GE. Application of a full potential method to supersonic aircraft design and analysis. *J Aircr* 1989;26:6–12.
- [214] Rose OJ, Miller DS, Pittman JL, Ashill PR, Fulker JL. Full-potential analysis of a supersonic delta wing/body. *J Aircr* 1989;26:235–40.
- [215] Shankar V, Osher S. An efficient, full potential, implicit method based on characteristics for supersonic flows. *AIAA J* 1983;21:1262–70.
- [216] Steger JL, Sorenson RL. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *J Comp Phys* 1979;33:405.
- [217] Miller D, Landrum E, Townsend J, Mason W. Pressure and force data for a flat wing and a warped conical wing having a shockless recompression at mach 1.62. *NASA TP* 1759, 1981.
- [218] Shankar V, Szema K-Y, Osher S. Treatment of supersonic flows with embedded subsonic regions. *AIAA J* 1985;23:41–8.
- [219] Ballhaus Jr W, Bridgeman J. Numerical solution techniques for unsteady transonic aerodynamic problems. *AGARD Special Course on Unsteady Aeronautics*, VKI, Rhode-St-Genese, Belgium: 1980.
- [220] van der Vooren J, Schippers H. Implicit finite-difference methods for the calculations on unsteady transonic potential flow about 2D airfoils — a discussion. In: Habashi W, editor. *Advances in Computer Transactions*. Swansea, UK: Pineridge Press Ltd., 1985. p. 185–213.
- [221] Engquist B, Majda A. Absorbing boundary conditions for the numerical simulation of waves. *Math Comp* 1977;31:629–51.
- [222] Engquist B, Majda A. Numerical radiation boundary conditions for unsteady transonic flow. *J Comp Phys* 1981;40:91–103.
- [223] Fung K. Far field boundary conditions for unsteady transonic flows. *AIAA J* 1981;19:180–3.
- [224] Kwak D. Nonreflecting far-field boundary conditions for unsteady transonic flow computation. *AIAA J* 1981;19:1401–7.
- [225] Fung K. A simple, accurate and efficient algorithm for unsteady transonic flow. In: Habashi W, editor. *Advances in Computational Transonics*. Swansea, UK: Pineridge Press Ltd., 1985. p. 317–39.
- [226] Ballhaus Jr WF, Goorjian PM. Computation of unsteady transonic flows by the indicial method. *AIAA J* 1978;16:117–24.
- [227] Rizzetta D. Time dependent response of a two dimensional airfoil in transonic flow. *AIAA J* 1979;17:26–32.
- [228] Rizzetta D, Chin W. Effect of frequency in unsteady transonic flow. *AIAA J* 1979;17:779–81.
- [229] Houwink R, van der Vooren J. Results of an improved version of LTRAN2 for computing unsteady airloads on airfoils oscillating intrasonic flow. *AIAA J* 1980;18:1008–10.
- [230] Guruswamy GP, Yang TY. Aeroelastic time response analysis of thin airfoils by transonic code LTRAN2. *Comp Fluids* 1981;9:409–25.
- [231] Edwards JW, Bennett RM, Whitlow Jr W, Seidel DA. Time-marching transonic flutter solutions including angle-of-attack effects. *J Aircr* 1983;20:899–906.
- [232] Traci RM, Albano ED, Farr JL. Small disturbance transonic flows about oscillating airfoils and planar wings. *AFFDL-TR-75-100*, 1975.
- [233] Isogai K. Numerical study of transonic flutter of a two-dimensional airfoil, TR-617T, National Aerospace Laboratory, Japan, 1980.
- [234] Caradonna FX, Isom MP. Numerical calculation of unsteady transonic potential flow over helicopter rotor blades. *AIAA J* 1976;14:482–8.

- [235] Borland C, Rizzetta D. Nonlinear transonic flutter analysis. *AIAA J* 1982;20:1606-15.
- [236] Borland CJ, Rizzetta DP. Transonic unsteady aerodynamics for aeroelastic applications, vol. I. Technical Development Summary, AFFDL-TR-80-3107, 1982.
- [237] Borland CJ, Rizzetta DP, Yoshihara H. Numerical solution of three-dimensional unsteady transonic flow over swept wings. *AIAA Paper* 80-1369, 1980.
- [238] Rizzetta D, Borland CJ. Unsteady transonic flow over wings including inviscid/viscous interaction. *AIAA J* 1983;21:363-71.
- [239] Isogai K, Suetsugu K. Numerical calculation of unsteady transonic potential flow over three-dimensional wings with oscillating control surfaces. *J Aircr* 1984;22:478-85.
- [240] Guruswamy P, Goorjian PM. Efficient algorithm for unsteady transonic aerodynamics of low-aspect-ratio wings. *J Aircr* 1985;22:193-9.
- [241] Guruswamy P, Goorjian PM. Unsteady transonic flow simulation on a full-span-wing-body configuration. *AIAA J* 1988;26:1450-6.
- [242] Guruswamy P, Goorjian PM, Tu EL. Unsteady transonics of a wing with tip store. *J Aircr* 1986;23:662-8.
- [243] Rodman LC, Nixon D, Huttshell LJ. Modifications to transonic flow codes for unsteady perturbations around an experimental mean. *J Aircr* 1989;26:765-71.
- [244] Batina JT. Unsteady transonic flow calculations for wing-fuselage configurations. *AIAA Paper* 86-0862-CP, 1986.
- [245] Batina JT. An efficient algorithm for solution of the unsteady transonic small-disturbance equation. *AIAA Paper* 87-0109, 1987.
- [246] Batina JT. Unsteady transonic algorithm improvements for realistic aircraft applications. *J Aircr* 1989;26:131-9.
- [247] Batina JT, Seidel DA, Bland SR, Bennett RM. Unsteady transonic flow calculations for realistic aircraft configurations. *J Aircr* 1989;26:21-8.
- [248] Gibbons MD, Batina JT. Supersonic far-field boundary conditions for transonic small-disturbance theory. *J Aircr* 1990;27:764-70.
- [249] Bennett RM, Bland SR, Batina JT, Gibbons MD, Mabey DG. Calculation of steady and unsteady pressures on wings at supersonic speeds with a transonic small-disturbance code. *J Aircr* 1991;28:175-80.
- [250] Goorjian PM, Guruswamy P. Unsteady transonic aerodynamic and aeroelastic calculations about airfoils and wings. *NASA TM* 85986, 1984.
- [251] Guruswamy GP. Interaction of fluids and structures for aircraft applications. *Comput Struct* 1988;10:1-13.
- [252] Guruswamy GP. Integrated approach for active coupling of structures and fluids. *AIAA J* 1989;27:788-93.
- [253] Guruswamy P, Goorjian PM. Comparisons between computations and experimental data in unsteady three-dimensional transonic aerodynamics, including aeroelastic applications. *AIAA Paper* 82-0690-CP, 1982.
- [254] Guruswamy P, Goorjian PM. Computations and aeroelastic applications of unsteady transonic aerodynamics about wings. *J Aircr* 1984;21:37-43.
- [255] Guruswamy P, Goorjian PM, Ide H, Miller GD. Transonic aeroelastic analysis of the B-1 wing. *J Aircr* 1986;23:547-53.
- [256] Guruswamy P, Goorjian PM, Tu EL. Transonic aeroelasticity of wings with tip stores. *J Aircr* 1987;25:688-95.
- [257] Guruswamy G, Marstiller J, Yang H, Goorjian P. Effects of viscosity and modes on transonic aerodynamics and aeroelastic characteristics of wings. *J Aircr* 1985;22:756-62.
- [258] Guruswamy GP, Tu EL. Effects of modal symmetry on transonic aeroelastic characteristics of wing-body configurations. *J Aircr* 1989;26:271-80.
- [259] Bennett RM, Batina JT, Cunningham HJ. Wing-flutter calculations with the CAP-TSD unsteady transonic small-disturbance program. *J Aircr* 1989;26:876-82.
- [260] Silva WA, Bennett RM. Application of transonic small disturbance theory to the active flexible wing model. *J Aircr* 1995;32:16-22.
- [261] Pitt DM, Fuglsang DF, Drouin DV. Application of XTRAN3S and CAP-TSD to fighter aircraft. *J Aircr* 1991;28:657-63.
- [262] Isogai K. Calculation of unsteady transonic flow over oscillating airfoils using the full potential equation. *AIAA Paper* 77-448, 1977.
- [263] Steger JL, Caradonna FX. A conservative implicit finite difference algorithm for the two dimensional unsteady transonic full potential equation. *NASA TM*-81211, 1980.
- [264] Shankar V, Ide H, Gorski J, Osher S. A fast, time-accurate, unsteady full potential scheme. *AIAA J* 1987;25:230-8.
- [265] Isogai K. A finite difference method for calculating unsteady transonic potential flow over 3D wings. In: Habashi W, editor. *Adv Comp Trans*. Swansea, UK: Pineridge Press Ltd., 1985. p. 289-315.
- [266] Bridgeman JO, Steger JL, Caradonna FX. A conservative finite difference algorithm for the unsteady transonic potential equation in generalized coordinates. *AIAA Paper* 82-1388, 1982.
- [267] Malone JB, Sankar NL, Sotomayer WA. Unsteady aerodynamic modeling of a fighter wing in transonic flow. *AIAA Paper* 84-1566, 1984.
- [268] Shankar V, Hiroshi I. Treatment of steady and unsteady three-dimensional flows using a time-accurate full potential scheme. *AIAA Paper* 85-4060, 1985.
- [269] Steinhoff J, Ramachandran K. Free-wake analysis of compressible rotor flows. *AIAA J* 1990;28:426-31.
- [270] Ramachandran K, Caradonna FX, Steinhoff J. The free-wake computation of rotor-body flows. *AIAA Paper* 90-1540, 1990.
- [271] Ramachandran K, Tung C, Caradonna FX. Rotor hover performance prediction using a free-wake, computational fluid dynamics method. *J Aircr* 1989;26:1105-10.
- [272] Bridgeman JO, Ramachandran K, Caradonna FX, Prichard D. The application of vorticity embedding to parallel blade-vortex interactions. *AIAA Paper* 94-1919, 1994.
- [273] Carradonna FX, Strawn RC, Bridgeman JO. An experimental and computational study of rotor-vortex interactions. 14th European Rotorcraft Forum, Milano, Italy, 1988.
- [274] Bridgeman JO, Prichard D, Caradonna FX. The development of a CFD potential method for the analysis of tilt-rotors. AHS Tech Spec Meeting on Rotorcraft Acoustics and Fluid Dynamics. Philadelphia, PA, 1991.

- [275] Strawn RC, Tung C. Prediction of unsteady transonic rotor loads with a full potential rotor code. 43rd Annual Forum of the AHS. St. Louis, MO, 1987.
- [276] Strawn RC, Caradonna FX. Conservative full-potential model for unsteady transonic rotor flows. *AIAA J* 1987;25:193–8.
- [277] Van Dalsem WR, Steger JL. The efficient simulation of separated three-dimensional viscous flows using the boundary layer equations. *AIAA Paper* 85-4064, 1985.
- [278] AGARD. Computational methods for aerodynamic design (inverse) and optimization, AGARD CP-463, Loen, Norway, 1989.
- [279] Dulikravich GS, editor. Proceedings of the Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III). Washington, DC, 1991.
- [280] Dulikravich GS. Aerodynamic shape design and optimization: status and trends. *J Aircr* 1992;29:1020–6.
- [281] Boerstol JW. Review of the application of hodograph theory to transonic aerofoil design and theoretical and experimental analysis of shock-free aerofoils. Symposium Transonicum II. Berlin: Springer, 1976.
- [282] Sobieczky H, Fung K-Y, Yu N, Seebass R. A new method for designing shock-free transonic configurations. *AIAA J* 1979;17:722–9.
- [283] Dulikravich DS, Sobieczky H. CAS22-FORTRAN program for fast design and analysis of shock-free airfoil cascades using fictitious-gas concept. NASA CR 3507, 1982.
- [284] Yu NJ. An efficient transonic shock-free wing redesign procedure using a fictitious gas method. *AIAA J* 1980;18:143–8.
- [285] Fung K, Sobieczky H, Seebass R. Shock-free wing design. *AIAA J* 1980;18:1153–8.
- [286] Raj P, Miranda LR, Seebass R. A cost-effective method for shock-free supercritical wing design. *J Aircr* 1982;19:283–9.
- [287] Steger JL, Klineberg JM. A finite-difference method for transonic airfoil design. *AIAA J* 1973;11:628–35.
- [288] Langley MJ. Numerical methods for two-dimensional and axisymmetric transonic flow, ARA Memo 143, 1973.
- [289] Tranen T. A rapid computer-aided transonic airfoil design method. *AIAA Paper* 74-501, 1974.
- [290] Volpe G, Melnik RE. The role of constraints in the inverse design problem for transonic airfoils. *AIAA Paper* 81-1233, 1981.
- [291] Volpe G. The inverse design of closed airfoils in transonic flow. *AIAA Paper* 83-504, 1983.
- [292] Shankar V, Malmuth ND, Cole JD. Computational transonic design procedure for wings and wing-body combinations. *AIAA Paper* 79-0344, 1979.
- [293] Shankar V, Malmuth ND, Cole JD. Computational transonic inverse procedure for wing design. *AIAA J* 1982;20:1044–50.
- [294] Henne P. Inverse transonic wing design method. *J Aircr* 1981;18:121–7.
- [295] Garabedian P, McFadden G. Design of supercritical swept wings. *AIAA J* 1982;20:289–91.
- [296] Slooff J. Computational procedures in transonic aerodynamic design. *Appl Comp Tran Aero*. AGARD-AG-266, 1982. p. 52–75.
- [297] Gally GA, Carlson LA. Transonic wing design using inverse methods in curvilinear coordinates. *J Aircr* 1988;25:1009–17.
- [298] Ratcliff RR, Carlson LA. Direct-inverse transonic wing-design method in curvilinear coordinates including viscous interaction. *J Aircr* 1991;28:803–4.
- [299] Takanashi S. Iterative three-dimensional transonic wing design using integral equations. *J Aircr* 1985;22:655–60.
- [300] Carlson LA, Weed RA. Direct-inverse transonic wing analysis-design method with viscous interaction. *J Aircr* 1986;23:711–8.
- [301] Malone JB, Vadyak J, Sankar LN. Inverse aerodynamic design method for aircraft components. *J Aircr* 1987;24:8–9.
- [302] Hassan AA, Charles BD. Airfoil design for helicopter rotor blades — a three-dimensional approach. *J Aircr* 1997;34:197–205.
- [303] Reuther J. Aerodynamic shape optimization using control theory. *RIACS Rep.* 96.09, 1996.
- [304] Hicks R, Henne P. Wing design by numerical optimization. *J Aircr* 1978;15:407–12.
- [305] Burgreen GW, Baysal O. Three-dimensional aerodynamic shape optimization of wings using sensitivity analysis. *AIAA Paper* 94-0094, 1994.
- [306] Luenberger DG. *Linear and Nonlinear Programming*, 2 [nd] Ed. Addison-Wesley Publishing Co., Inc., Reading, MA 1984;2:2.
- [307] Vanderplaats GM. CONMIN — A FORTRAN program for constrained function minimization. NASA TM X-62282, 1973.
- [308] Gill PE, Murray WM. Quasi-Newton methods for unconstrained optimization. *J Inst Math Appl* 1972;9:91–108.
- [309] Gill P, Murray W, Saunders M, Wright M. User's guide for NPSOL (Version 4.0): A FORTRAN package for nonlinear programming. Stanford University Technical Report SOL86-2, 1986.
- [310] Vanderplaats GM. ADS — A FORTRAN program for automated design synthesis, Version 3.0, Engineering Design Optimization, Inc., Santa Barbara, CA, 1988.
- [311] Cheung S. Aerodynamic design: Parallel CFD and optimization routines. *AIAA Paper* 95-1748, 1995.
- [312] Cheung S. Parallel computational fluid dynamics design on network-based computer. *J Aircr* 1996;33:561–6.
- [313] Melvin R, Huffman W, Young D, Johnson F, Hilmes C, Bieterman M. Recent progress in aerodynamic design optimization. Boeing Info. & Support Services, ISSTECH-96-024, 1996.
- [314] Hicks RM, Murman EM, Vanderplaats GN. An assessment of airfoil design by numerical optimization. NASA TM X-3092, 1974.
- [315] Kennelly Jr RA. Improved method for transonic airfoil design-by-optimization. *AIAA Paper* 83-1864, 1983.
- [316] Ghielmi L, Marazzi R, Baron A. A tool for automatic design of airfoils in different operating conditions. Paper 18, AGARD CP-463, 1990.
- [317] Haney HP, Johnson RR, Hicks RM. Computational optimization and wind tunnel test of transonic wing designs. *J Aircr* 1980;17:457–63.
- [318] Hicks RM. Transonic wing design using potential flow codes — successes and failures. *SAE Paper* 810565, 1981.

- [319] George MW, Wedan BW, Johnson RR. Computational wing design for an advanced trainer. AIAA Paper 83-0093, 1983.
- [320] Destarac D, Reneaux J, Gisquet D. Optimisation Numerique de Voilures en Regime Transsonique. Paper 21, AGARD CP 463, 1990.
- [321] Reneaux J, Allongue M. Definitions de profils et de pales d'Helicoptere par Optimisation Numerique, Paper 19, AGARD CP-463, 1990.
- [322] Aidala PV. Transonic fighter design using numerical optimization. ICAS Paper 1980;80:2.1.
- [323] El-banna HM, Carlson LA. Determination of aerodynamic sensitivity coefficients based on the transonic small perturbation formulation. AIAA J 1990;27:507–15.
- [324] El-banna HM, Carlson LA. Aerodynamic sensitivity coefficients using the three-dimensional full potential equation. J Aircr 1994;31:1071–7.
- [325] Arslan AE, Carlson LA. Integrated determination of sensitivity derivatives for an aeroelastic transonic wing. J Aircr 1996;33:224–31.
- [326] Jameson A. Aerodynamic design via control theory. J Sci Comp 1988;3:233–60.
- [327] Jameson A. Automatic design of transonic airfoils to reduce the shock induced pressure drag. 31st Israel Annual Conference on Aviation and Aeronautics. Tel Aviv, 1990. p. 5–17.
- [328] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989. p. 59–88.
- [329] Schwefel HP. Numerische Optimierung von Computermodellen mittels der Evolutionsstrategie. Basel/Stuttgart: Birkhäuser-Verlag, 1977.
- [330] Davis L. Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991.
- [331] Quagliarella D, Della Cioppa A. Genetic algorithms applied to the aerodynamic design of transonic airfoils. AIAA Paper 94-1896-CP, 1994.
- [332] Vicini A, Quagliarella D. Inverse and direct airfoil design using a multiobjective genetic algorithm. AIAA J 1997;35:1499–505.
- [333] Obayashi S, Yamaguchi Y, Nakamura T. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform. J Aircr 1997;34:690–3.
- [334] Obayashi S, Tsukahara T. Comparison of optimization algorithms for aerodynamic shape design. AIAA J 1997;35:1413–5.
- [335] Bock K-W. Aerodynamic design by optimization. Paper 20, AGARD CP-463, 1990.
- [336] Atta EH. Component-adaptive grid interfacing. AIAA Paper 81-0382, 1981.
- [337] Lê TH. Transonic potential flow calculation about complex bodies by a technique of overlapping subdomains, Eighth International Conference on Numerical Methods in Fluid Dynamics. ONERA TP 1982-58, 1982.
- [338] Atta EH, Vadyak J. A grid overlapping scheme for flow-field computations about multicomponent configurations. AIAA J 1983;21:1271–7.
- [339] Steger JL, Dougherty FC, Benek JA. A chimera grid scheme. In: Ghia K, Ghia U, editors. Advances in Grid Generations. ASME FED-5. 1983. p. 59–69.
- [340] Benek JA, Steger JL, Dougherty FC. A flexible grid embedding technique with application to the Euler equations. AIAA Paper 83-1944, 1983.
- [341] Dougherty FC, Benek JA, Steger JL. On application of chimera grid scheme to store separation to viscous flows. NASA TM 88193, 1985.
- [342] Epstein B, Luntz AL, Nachshon A. Multigrid transonic computations about arbitrary aircraft configurations. J Aircr 1989;26:751–9.
- [343] Lifshitz YB, Solntsev IA, Velichko SA. Analysis of transonic flow by method of viscous-inviscid interaction: Correction for wind tunnel wall interference. AIAA Paper 96-1981, 1996.
- [344] Lifshitz YB, Velichko SA. AEROFOIL — A computer code library for numerical analysis of viscous transonic flow around a wing section. Rech Aerospaciale, 1995. p. 73–83.
- [345] Holst TL. A chimera donor cell search algorithm suitable for solving the full potential equation. AIAA Paper 97-2259, 1997.
- [346] Holst TL. Fast aerodynamic analysis suitable for the preliminary design environment. AIAA Paper 99-3133, 1999.
- [347] Sankar NL, Bharadvaj BK, Tsung F-L. Three-dimensional Navier-Stokes/full-potential coupled analysis for viscous transonic flow. AIAA J 1993;31:1857–62.
- [348] Bangalore AK, Moulton MA, Caradonna FX. The development of an overset/hybrid method for rotorcraft applications. AHS Technical Special Meeting for Rotorcraft Aeronautics and Aeroacoustics, Williamsburg, VA, 1997.
- [349] Berkman ME, Sankar LN, Berezin CR, Torok MS. Navier-Stokes/full potential/free-wake method for rotor flows. J Aircr 1997;34:635–40.
- [350] Moulton MA, Bridgeman JO, Caradonna F. Development of an overset/hybrid CFD method for the prediction of hovering performance. 53rd AHS Annual Forum, Virginia Beach, VA, 1997.
- [351] Moulton MA, Ramachandran K, Hafez MM, Caradonna F. The development of a hybrid CFD method for the prediction of hover performance. AHS Specialists' Meeting on Aeromechanics Technology and Product Design for the 21st Century, Bridgeport, CT, 1995.
- [352] Steger JL, Rizk YM. Generation of three-dimensional body-fitted coordinates using hyperbolic partial differential equations. NASA TM-86753, 1985.
- [353] Chan WM, Chiu I, Buning PG. User's manual for the HYPGEN hyperbolic grid generator and the HGUI graphical user interface. NASA TM 108791, 1993.
- [354] Glowinski R, Periaux J, Pironneau O. Transonic flow simulation by the finite-element method via optimal control. Finite Elements in Fluids 1976;3:249–59.
- [355] Ecer A, Akay HU. Application of finite element method for the solution of transonic flow. Finite Elements in Fluids 1976;3:191–201.
- [356] Vigneron Y, Brocard O, Bousquet J, Lejal T. Une Methode Variationelle d'Elements Finis Pour la Resolution d'Ecoulements Transoniques Tridimensionnels. AGARD CP 285, 1980.
- [357] Bristeau M, Glowinski R, Periaux J, Perrier P, Pironneau O, Poirier G. Applications of optimal control and finite