Cartesian Mesh Solution for Axisymmetric Transonic Potential Flow Around Inlets

T. A. Reyhner*
Boeing Commercial Airplane Company, Seattle, Wash.

A procedure has been developed for computing transonic solutions of the exact axisymmetric compressible potential flow equation for flow about inlets. A cartesian mesh is used in physical coordinates to eliminate restrictions on the geometries that can be treated. The method is shown to be fast and stable; and agreement between analysis and data is excellent. The solution procedure is valid for any geometry or any number of surfaces. The method was developed as a first step toward a general three-dimensional analysis and all parts of the procedure apply directly to three-dimensional flow computation with a minimum of further development.

Nomenclature

A_∞ = capture area
A_{HI} = area enclosed by Hilite
A_{TH} = throat area
a = speed of sound
n = coordinate normal to surface
$\boldsymbol{n}$ = unit normal to surface
n_r, n_z = components of $\boldsymbol{n}$
q = velocity, $(\phi_r{}^2 + \phi_z{}^2)^{1/2}$
r = radius
Δr = step size in r
S = surface point
s = coordinate along surface
z = axial coordinate
Δz = step size in z
γ = ratio of specific heats
λ = eigenvalue
ϕ = potential function
ϕ_n = $\partial\phi/\partial n$, velocity normal to surface
ϕ_r = $\partial\phi/\partial r$, radial velocity
ϕ_z = $\partial\phi/\partial z$, axial velocity
ω = local over-relaxation parameter
ω_n = overall (nominal) over-relaxation parameter

Superscripts

$+$ = value after updating

Subscripts

m,n = mesh indices
$\ell, \ell\ell, u$ = step-size subscripts (Fig. 3)
r,z = partial derivatives
∞ = freestream

Introduction

IN the last few years significant progress has been made on the problem of analytic solutions of transonic flow problems. The work of Murman and Cole,[1] which treats the small-disturbance form of the potential equation, but allows shocks, used upwind differencing in supersonic flow regions to allow the computation of mixed elliptic and hyperbolic equation systems. Another approach, the integration of the equations of unsteady compressible flow forward in time to approach a steady state, developed by Magnus and Yoshihara,[2] has shown promising results, but the computations are very lengthy. The Murman and Cole approach of solving the steady form of the small-disturbance equation using relaxation has been extended by many investigators. Solutions available include small-disturbance theory applied to three-dimensional bodies,[3] and a solution of the exact three-dimensional potential equation for a class of wing planforms.[4] Most of these works have involved transformations such that the body surface coincides with a coordinate surface or small-disturbance theory. Carlson[5] has used cartesian coordinates to calculate the flow about two-dimensional airfoils. Solutions for flow about axisymmetric inlets include those by Arlinger,[6] Colehour,[7] and Caughey and Jameson.[8]

One of the critical points in the performance envelope of an airplane jet-engine inlet is the low speed, high angle-of-attack, high airflow condition that occurs just after takeoff. To meet angle-of-attack performance criteria without a significant increase in cruise drag, inlets are often designed with a different lip shape on the bottom quadrant than on the top. As a consequence many practical inlets are not axisymmetric. An additional problem is created by nacelles that are mounted close to the wing, because the interaction between the wing, strut, and nacelle becomes important. The above flows are all three-dimensional, and are mentioned in order of the increased geometrical complexity. These are very practical problems; their existence makes the development of a three-dimensional potential flow program with good geometrical flexibility very desirable. This paper describes a step towards that goal.

It is obvious that a solution of the exact three-dimensional potential flow equation using a cartesian mesh would have many advantages. A difficulty with programs using body-oriented coordinate systems is that they are generally restricted in the number of different geometries they are capable of handling. It was decided that the necessary differencing and solution procedures for a three-dimensional cartesian mesh solution could be developed faster and with less work by studying a two-dimensional or an axisymmetric problem. The axisymmetric problem was chosen because it applied to inlets, and analytic and test results were available to guide the work.

An analysis and computer program have been developed to solve the exact compressible potential flow equation for transonic axisymmetric flow about an arbitrarily shaped cowl with or without a centerbody. This analysis uses a cartesian mesh consisting of the radial coordinate r and the axial coordinate z. The analysis is fast, reliable and accurate. Equally valid solutions of the axisymmetric flow problem[6-9] are presently available, but the approaches used do not extend directly to three-dimensions and, depending on the approach,

Received July 7, 1976; presented as Paper 76-421 at the AIAA 9th Fluid and Plasma Dynamics Conference, San Diego, Calif., July 14-16, 1976; revision received Feb. 7, 1977.

Index categories: Aerodynamics; Subsonic and Transonic Flow.

*Specialist Engineer, Propulsion Technology Research. Associate Fellow AIAA.

are oriented to a narrow class of geometries. This approach directly extends to three-dimensional flows; the only apparent limitation in three-dimensions would be the need for sufficient mesh lines to resolve body features. The analysis does not use any transformations and is thus not limited by body shape. This program would easily extend to multiple surfaces in the flowfield, although, at present no provision has been made for lifting surfaces (Kutta condition).

Approach

The approach is to use finite-difference formulas written for the cartesian mesh nodes plus the additional nodes created by the intersection of the mesh lines and the surfaces. Mesh nodes that are interior to surfaces are not used. A variable spacing of mesh lines is used in order to concentrate mesh nodes around body features that would otherwise be difficult to resolve. Solution of the difference equations is by line relaxation. The primary difficulties with this approach are the bookkeeping problems associated with which mesh points are adjacent to which surface points, and of keeping track and generating difference formulas for the many different kinds of "computational stars" that can exist in the vicinity of surfaces.

Equation and Boundary Conditions

The equation to be solved is the exact equation for inviscid, irrotational flow formulated in terms of a velocity potential. This form of the equation has the advantage that there is only one dependent variable that has to be stored at each mesh node. This is particularly important in any extension to three dimensions. At mesh nodes the exact compressible potential flow equation in cylindrical coordinates is satisfied.

$$(a^2-\phi_r{}^2)\phi_{rr}+(a^2-\phi_z{}^2)\phi_{zz}-2\phi_r\phi_z\phi_{rz}+\frac{a^2}{r}\phi_r=0 \qquad (1)$$

where

$$a^2=a_\infty^2-\frac{\gamma-1}{2}(\phi_r{}^2+\phi_z{}^2-q_\infty{}^2) \qquad (2)$$

The boundary condition on the surface of the cowl and on the centerbody is that the velocity normal to the surface, ϕ_n, is zero. On the axis, $r=0$, the differential equation becomes[8]

$$2a^2\phi_{rr}+(a^2-\phi_z{}^2)\phi_{zz}=0 \qquad (3)$$

For inlet flowfield calculations it is adequate to compute the flowfield in a cylinder much larger than the inlet. The boundary conditions on the cylinder are no flow through the cylinder walls, ϕ is specified on one end, and ϕ_n is specified on the other end. The computational region is shown in Fig. 1. This type of boundary condition approximates an inlet in a very large wind tunnel with a circular cross section.

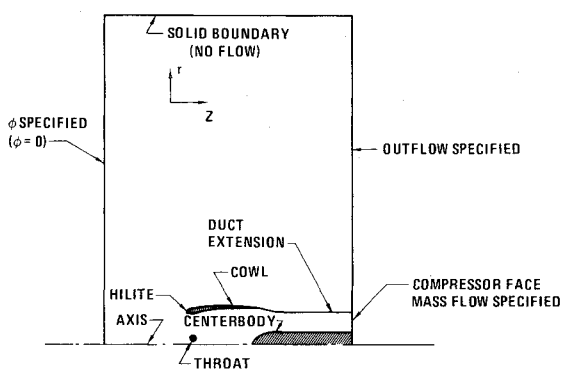

Fig. 1 Standard computation region.

Table 1 Types of mesh points

Type	Description
0	Regular mesh point
−1	Interior of surface
1	Upper or lower boundary point
−2	Surface affected – upper right unavailable
−3	Surface affected – upper left unavailable
−4	Surface affected – lower left unavailable
−5	Surface affected – lower right unavailable
2	Surface adjacent – body up
3	Surface adjacent – body down
5	Surface adjacent – body right
7	Surface adjacent – body left
7×3=21	Surface adjacent – body left and down
7×2=14	Surface adjacent – body left and up
3×5=15	Surface adjacent – body right and down
2×5=10	Surface adjacent – body right and up

Solution Procedure

Solution of the partial-differential equations is by substituting difference quotients for partial derivatives to form finite-difference equations. The resulting set of difference equations is highly nonlinear. The difference equations are solved by a line-relaxation technique, including overrelaxation. The subject of relaxation is quite complex and will not be discussed here. Reference 10 is a recent paper that can provide further information on relaxation methods applied to this type of problem. Early in this work, a point-relaxation scheme was tried, but it suffered from extremely slow convergence in parts of the duct where $\Delta z/\Delta r$ was very large. This is a severe handicap because enforcing the compressor-face mass-flow boundary condition is critical to predicting inlet performance. Line relaxation was used only on vertical lines, although in principle it could be used on horizontal lines as well.

Point Types and Mesh Bookkeeping

One of the primary problems associated with a cartesian mesh is the bookkeeping task of associating surface points with their adjacent mesh points. The scheme used is based on cataloging the different types of mesh points, associating an integer with each general type of mesh point, and storing these integers. The catalog of mesh points is given in Table 1. Representative point types for the catalog are shown in Fig. 2. For the computer program, two arrays are required of size $M\times N$ (number of $z=$ constant mesh lines times the number of $r=$ constant mesh lines). The arrays are indexed with the m and n counters for the mesh lines. One array carries the value of the potential function ϕ at the mesh nodes. Points interior to the surfaces are not used. A second integer array contains either the type of the mesh point or an array index. If the integer is less than or equal to one it is the point type. If it is greater than one it is recognized to be the index for shorter arrays (of the order of the number of surface points) which

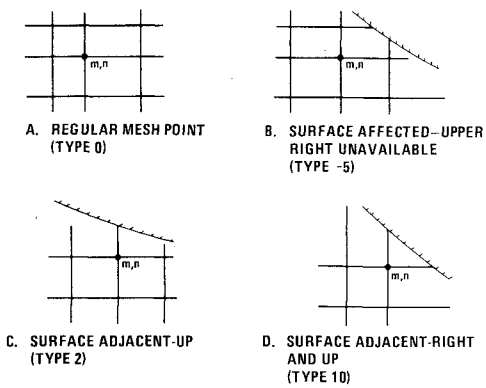

Fig. 2 Representative point types.

contain the type of the mesh point, and the index or indices of the surface point or points adjacent to the mesh point. Surface points are indexed sequentially along the surfaces. Associated with the surface points, and indexed with the surface point number, are arrays containing the potential function ϕ, the r and z coordinates of the surface point, and the components of the surface normal, n_r and n_z.

The individual parts of the computer program, such as the differencing for one type of mesh point, are relatively simple and straightforward. The overall complexity of the computer program is due to the number of point types, the logic for handling the switching between the different point types, and the possibility of central or upwind differencing, or in the transonic regime a combination of the two, being used.

Finite-Difference Formulas

At regular mesh points the difference formulas are standard ones. The quantities ϕ_r and ϕ_z are calculated using central difference formulas that are formally of second order. A typical example (see Fig. 3 for notation) is

$$\phi_z|_{m,n} = \frac{\Delta z_\ell{}^2 \phi_{m+1,n} + (\Delta z_u{}^2 - \Delta z_\ell{}^2)\phi_{m,n} - \Delta z_u{}^2 \phi_{m-1,n}}{\Delta z_\ell \Delta z_u (\Delta z_\ell + \Delta z_u)} \tag{4}$$

ϕ_r and ϕ_z are calculated using all old values (values from the previous sweep). The difference quotients used for the terms ϕ_{rr}, ϕ_{zz}, and ϕ_{rz} depend on whether the flow is subsonic or transonic. If the flow is subsonic ϕ_{rr} and ϕ_{zz} are calculated using central differences

$$\phi_{rr} = \frac{\Delta r_\ell \phi_{m,n+1}{}^+ - (\Delta r_\ell + \Delta r_u)\phi_{m,n}{}^+ + \Delta r_u \phi_{m,n-1}{}^+}{\tfrac{1}{2}\Delta r_\ell \Delta r_u (\Delta r_\ell + \Delta r_u)} \tag{5}$$

$$\phi_{zz} = \frac{\Delta z_\ell \phi_{m+1,n} - (\Delta z_\ell + \Delta z_u)\left(\frac{1}{\omega}\phi_{m,n}{}^+ + \left(1 - \frac{1}{\omega}\right)\phi_{m,n}\right) + \Delta z_u \phi_{m-1,n}{}^+}{\tfrac{1}{2}\Delta z_\ell \Delta z_u (\Delta z_\ell + \Delta z_u)} \tag{6}$$

Since line relaxation along vertical lines is being used, the expression for ϕ_{rr} is evaluated using all new (calculated during current sweep) values. The expression for ϕ_{zz} involves, by necessity, a combination of old and new values. To accelerate convergence it also includes an over-relaxation parameter, ω. The sweeping direction is with the primary flow (left to right), thus a new value for ϕ is available at $(m-1, n)$ and convergence is faster if it is used. The over-relaxation parameter is computed as follows

$$\omega = 1 + \left(\frac{\Delta z_\ell + \Delta z_u}{\text{maximum}(\Delta z_\ell, \Delta z_u)} - 1\right)(\omega_n - 1) \tag{7}$$

$$0 < \omega_n < 2$$

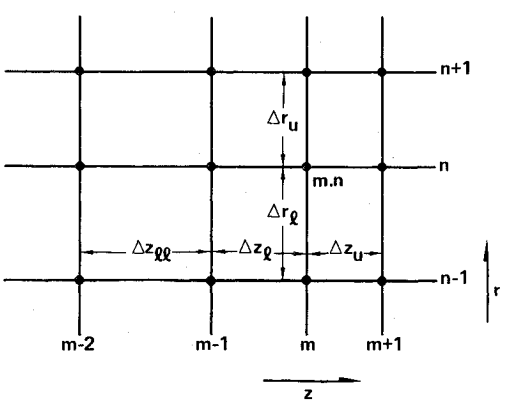

Fig. 3 Regular mesh node (m,n).

If $\Delta z_\ell = \Delta z_u$, $\omega = \omega_n$, but if $\Delta z_\ell \neq \Delta z_u$, ω is reduced. This is necessary to insure that the coefficient of $\phi_{m,n}^+$ is always greater than or equal to the coefficients of $\phi_{m-1,n}^+$ or $\phi_{m+1,n}$. If these were the only other terms in the equation this condition would be strictly required for stability. Since there are other terms in the equations it is not strictly required, but it is a simple way to ensure stability while allowing ω_n to be almost equal to 2 (1.85 – 1.95 used in practice). The use of large values for ω_n greatly speeds convergence.

For subsonic flow ϕ_{rz} is differenced using all old values

$$\phi_{rz} = \frac{\Delta z_\ell{}^2 \phi_r|_{m+1,n} + (\Delta z_u{}^2 - \Delta z_\ell{}^2)\phi_r|_{m,n} - \Delta z_u{}^2 \phi_r|_{m-1,n}}{\Delta z_\ell \Delta z_u (\Delta z_\ell + \Delta z_u)} \tag{8a}$$

$$\phi_r|_{k,n} = \frac{\Delta r_\ell{}^2 \phi_{k,n+1} + (\Delta r_u{}^2 - \Delta r_\ell{}^2)\phi_{k,n} - \Delta r_u{}^2 \phi_{k,n-1}}{\Delta r_\ell \Delta r_u (\Delta r_\ell + \Delta r_u)} \tag{8b}$$

This formula is formally of second order.

The term $a^2\phi_r/r$ is differenced using

$$\phi_r = \frac{\Delta r_\ell{}^2 \phi_{m,n+1}{}^+ + (\Delta r_u{}^2 - \Delta r_\ell{}^2)\phi_{m,n}{}^+ - \Delta r_u{}^2 \phi_{m,n-1}{}^+}{\Delta r_\ell \Delta r_u (\Delta r_\ell + \Delta r_u)} \tag{9}$$

It is expected that this term could also be calculated using all old values without any significant changes in the behavior of the procedure.

Upwind Differencing

Progress in transonic potential flow calculation depended on the development of a technique to handle supersonic flow as the above formulas lead to unstable calculations in regions where the Mach number is greater than one. Murman and Cole[1] found that by using upwind differencing for the second derivative in the streamwise direction the calculation was stabilized. The original work using upwind differencing was for small-disturbance theory where the flow was almost aligned with one of the coordinates. South and Jameson[9] have extended the use of upwind differencing to the exact potential equation for flows which are not necessarily aligned with the coordinate system. They developed a rotated difference scheme where the partial differential equation was, for Mach numbers greater than one, divided into two parts. The terms in one part were differenced using central differences, and the terms in the other part were differenced using upwind differences. The scheme used here differs from South and Jameson's in that every term in the equation is treated separately, and the differencing of each term is switched independently from central to upwind differencing depending on certain criteria. The ideas of Jameson[11] have been used in developing the mix between old and new values in the scheme. The upwind difference formula for ϕ_{zz} when $\phi_z > a$ is

$$\phi_{zz} = \frac{(\Delta z_{\ell\ell} + \Delta z_\ell)(\phi_{m,n}{}^+ - \phi_{m-1,n}{}^+) - \Delta z_\ell(\phi_{m,n} - \phi_{m-2,n})}{\tfrac{1}{2}\Delta z_{\ell\ell} \Delta z_\ell (\Delta z_\ell + \Delta z_{\ell\ell})} \tag{10}$$

This formula is used when $\phi_z{}^2 > a^2$. It is formally of first order. The expressions for ϕ_{rr} used when $\phi_r{}^2 > a^2$ are similar. There is one expression for the upwind difference for ϕ_{rr} if $\phi_r > a$ and another for $\phi_r < -a$.

The reasoning given by Jameson for using this combination of old and new values of ϕ in the difference formulas is based on looking at the relaxation process as marching in an artificial time coordinate. A simpler explanation of why this scheme behaves quite well is as follows. First note that with this weighting the coefficient of the unknown $\phi_{m,n}{}^{+}$ is greater than or equal to the coefficients of other terms in the formulas. This insures that any errors in the ϕ values for other terms in the equation do not get amplified when they contribute to the calculation of $\phi_{m,n}{}^{+}$. Second note that any change in $\phi, \Delta\phi$, originating upstream during this sweep is propagated downstream unchanged. This essentially means that corrections upstream are quickly propagated downstream without being diminished or the sign changed. If one looks at the effects of the over-relaxation parameter for the subsonic differencing it can be seen to be very similar.

The upwind difference formula for ϕ_{rz}, $\phi_r > 0$ and $\phi_z > 0$ is

$$\phi_{rz} = \frac{\phi_{m,n}{}^{+} - \phi_{m,n-1}{}^{+} - \phi_{m-1,n}{}^{+} + \phi_{m-1,n-1}{}^{+}}{\Delta r_\ell \Delta z_\ell} \tag{11}$$

The upwind formula for ϕ_{rz} is used when $\phi_r{}^2 > a^2$ or $\phi_z{}^2 > a^2$. If $M > 1$ but $\phi_r{}^2$ and $\phi_z{}^2$ are less than a^2 then a combination of the upwind and central difference formulas are used depending on the parameter

$$\theta = \frac{(a^2 - \phi_r{}^2)(a^2 - \phi_z{}^2)}{\phi_r{}^2 \phi_z{}^2} \tag{12}$$

$$\phi_{rz} = \theta \phi_{rz}|_{\text{central diff.}} + (1-\theta)\phi_{rz}|_{\text{upwind diff.}} \tag{13}$$

This parameter equals one for $M = 1$; thus it gives a smooth transition between the central and upwind difference formulas. It can also be shown that this switching insures that the coefficient of $\phi_{m,n}{}^{+}$ remains larger than that of other terms in the difference equation for any flow angle with respect to the coordinate system.

Differencing Near Surfaces

The only change required in the central difference formulas for ϕ_{rr} and ϕ_{zz} [(Eqs. (5) and (6)] for surface-adjacent points is to replace the regular mesh point with a surface point in the formula and to use the step size to the surface. Complications arise in the calculation of ϕ_{rz}, and in upwind differencing for ϕ_{rr} and ϕ_{zz} if the first point upwind is a surface point. A couple of examples will be given, but space prohibits a complete catalog of formulas.

The first example is the computation of ϕ_{rz} for the configuration of Fig. 4a. Different formulas are used if the surface slope is greater or less than 45°. For a horizontal type (<45°) surface the formula is

$$\phi_{rz} = \frac{\Delta z_\ell{}^2 \phi_{r|_S} + (\Delta z_u{}^2 - \Delta z_\ell{}^2)\phi_{r|_{m,n}} - \Delta z_u{}^2 \phi_{r|_{m-1,n}}}{\Delta z_\ell \Delta z_u (\Delta z_\ell + \Delta z_u)} \tag{14}$$

$$\phi_{r|_S} = \frac{((\Delta r_1 + \Delta r_2)^2 - \Delta r_1{}^2)\phi_S - (\Delta r_1 + \Delta r_2)^2 \phi_A + \Delta r_1{}^2 \phi_B}{\Delta r_1 \Delta r_2 (\Delta r_1 + \Delta r_2)} \tag{15}$$

The values of ϕ_A and ϕ_B are calculated using second-order interpolation along the $n-1$ and $n-2$ mesh lines respectively. $\phi_{r|_{m,n}}$ and $\phi_{r|_{m-1,n}}$ are computed normally. If upwind differencing is required and $\phi_r < 0$, $\phi_C{}^{+}$ would be calculated using interpolation and then a straightforward adaption of Eq. (11) would be used.

The second example is the computation of ϕ_{zz} upwind at (m,n) for the configuration of Fig. 4b.

$$\phi_{zz} = \frac{\dfrac{\phi_{m,n}{}^{+} - \phi_S{}^{+}}{\Delta z_\ell} - \phi_z|_S}{\Delta z_\ell / 2} \tag{16}$$

$$\phi_z|_S = -n_r \phi_s|_S = |n_r|(C_1 \phi_S{}^{+} + C_2) \tag{17}$$

where C_1 and C_2 are computed using second-order upwind differencing along the surface. With the proper mix of old and new values

$$\phi_{zz} = \frac{2}{\Delta z_\ell{}^2}[(1 + |n_r|\Delta z_\ell C_1)(\phi_{m,n}{}^{+} - \phi_S{}^{+}) - |n_r|\Delta z_\ell (C_1 \phi_{m,n} + C_2)] \tag{18}$$

where C_2 is calculated using all old values.

In the vicinity of the surface, additional problems can develop because of the possibilities of very irregular mesh spacing. This is particularly a problem when the step size is very small. Referring to Fig. 5 as an example, if $\Delta_{z_u} << \Delta_{z_\ell}$

$$\phi_z{}^{(1)} = \frac{\Delta z_\ell{}^2 \phi_S + (\Delta z_u{}^2 - \Delta z_\ell{}^2)\phi_{m,n} - \Delta z_u{}^2 \phi_{m-1,n}}{\Delta z_\ell \Delta z_u (\Delta z_\ell + \Delta z_u)} \tag{19a}$$

$$\approx \frac{\phi_S - \phi_{m,n}}{\Delta z_u} \tag{19b}$$

and if ϕ_S is in error by an amount ϵ the error in ϕ_z is of order $\epsilon/\Delta z_u$ which can be arbitrarily large. To compensate for this problem alternate formulas were developed. For the sample above if $\phi_z^{(1)}$ is the usual formula, then $\phi_z^{(2)}$ is defined as

$$\phi_z{}^{(2)} = \frac{(\Delta z_T{}^2 - \Delta z_\ell{}^2)\phi_S - (\Delta z_T{}^2 - \Delta z_u{}^2)\phi_{m-1,n} + (\Delta z_\ell{}^2 - \Delta z_u{}^2)\phi_{m-2,n}}{\Delta z_{\ell\ell}(\Delta z_u + \Delta z_\ell)(\Delta z_u + \Delta z_T)} \tag{20}$$

$$\Delta z_T = \Delta z_\ell + \Delta z_{\ell\ell}$$

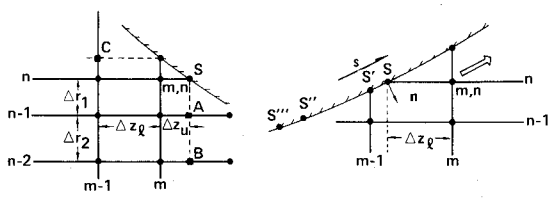

Fig. 4 Examples of differencing for surface adjacent points. a) (m,n) surface adjacent on two sides. b) $\phi_z > a$ and first upwind point is a surface point.

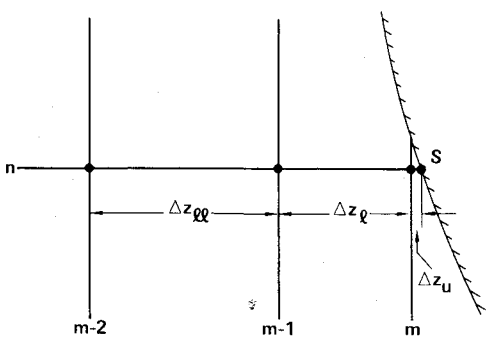

Fig. 5 Irregular spacing near surface.

and ϕ_z is defined as

$$\phi_z = \frac{2\Delta z_u}{\Delta z_\ell}\,\phi_z{}^{(1)} + \left(1 - \frac{2\Delta z_u}{\Delta z_\ell}\right)\phi_z{}^{(2)} \tag{21}$$

for $\Delta z_u < \frac{1}{2}\Delta z_\ell$. Both formulas are of second order. This type of weighting was applied wherever ϕ_z or ϕ_r were calculated adjacent to a surface and the step size to the surface was less than half of the regular mesh step size.

As a further effort to minimize the effects of very small step sizes at surface intersects, the difference equation for the potential flow has been weighted with a second-order interpolation equation near the surface. As an example for the configuration of Fig. 5 the interpolation equation is

$$\phi_{m,n}{}^{+} = \phi_{m-1,n}{}^{+} + \frac{\Delta z_\ell(\Delta z_\ell + \Delta z_{\ell\ell})}{(\Delta z_u + \Delta z_\ell)(\Delta z_u + \Delta z_\ell + \Delta z_{\ell\ell})}(\phi_S{}^{+} - \phi_{m-1,n}{}^{+}) - \frac{\Delta z_\ell \Delta z_u}{\Delta z_{\ell\ell}(\Delta z_u + \Delta z_\ell + \Delta z_{\ell\ell})}(\phi_{m-2,n}{}^{+} - \phi_{m-1,n}{}^{+}) \tag{22}$$

If $\Delta z_u < \frac{1}{2}\Delta z_\ell$ then the equation for $\phi_{m,n}{}^{+}$ becomes

$$\frac{2\Delta z_u}{\Delta z_\ell}(\text{Regular equation for } \phi_{m,n}{}^{+} \text{ normalized so that the coefficient of } \phi_{m,n}{}^{+} = 1) + \left(1 - \frac{2\Delta z_u}{\Delta z_\ell}\right)[\text{Eq. (22)}] \tag{23}$$

Surface-Point Boundary-Condition Difference Formulas

The preceeding discussion has dealt with the generation of a difference equation for each of the mesh nodes inside of the flowfield. For the nodes created by the intersection of a mesh line with a surface, the difference equation is a representation of the boundary condition

$$\phi_n = \boldsymbol{n}\cdot\text{grad}\,\phi = n_r\phi_r + n_z\phi_z = 0 \tag{24}$$

Except for the rare surface node that is also an intersection of regular mesh lines, either ϕ_r or ϕ_z is calculated using points computed by interpolation. For the example of a r = constant mesh line intersect, S, shown in Fig. 6 the equations are as follows

$$\phi_z{}^{(1)} = -i\frac{(\Delta z_{12}{}^2 - \Delta z_1{}^2)\phi_S{}^{+} - \Delta z_{12}{}^2\phi_{m,n}{}^{+} + \Delta z_1{}^2\phi_{m+i,n}{}^{+}}{\Delta z_1 \Delta z_2 \Delta z_{12}} \tag{25}$$

$$\phi_z{}^{(2)} = -i\frac{(\Delta z_{123}{}^2 - \Delta z_{12}{}^2)\phi_S{}^{+} - \Delta z_{123}{}^2\phi_{m+i,n}{}^{+} + \Delta z_{12}{}^2\phi_{m+2i,n}{}^{+}}{\Delta z_{12}\Delta z_3\Delta z_{123}} \tag{26}$$

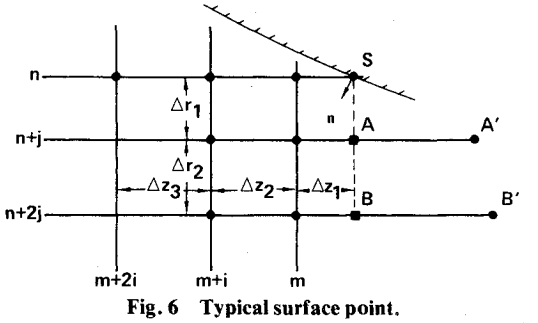

Fig. 6 Typical surface point.

$$\Delta z_{12} = \Delta z_1 + \Delta z_2, \quad \Delta z_{123} = \Delta z_1 + \Delta z_2 + \Delta z_3$$

$$\phi_z = \beta\phi_z{}^{(1)} + (1-\beta)\phi_z{}^{(2)} \tag{27}$$

$$\beta = \begin{cases} 2\Delta z_1/\Delta z_2 & \Delta z_1 < \frac{1}{2}\Delta z_2 \\ 0 & \Delta z_1 \geq \frac{1}{2}\Delta z_2 \end{cases}$$

$$\phi_r = -J \times \frac{((\Delta r_1 + \Delta r_2)^2 - \Delta r_1{}^2)\phi_S{}^{+} - (\Delta r_1 + \Delta r_2)^2\phi_A{}^{+} + \Delta r_1{}^2\phi_B{}^{+}}{\Delta r_1 \Delta r_2(\Delta r_1 + \Delta r_2)} \tag{28}$$

$\phi_A{}^{+}$ and $\phi_B{}^{+}$ are calculated using second-order interpolation along the $n+j$ and $n+2j$ mesh lines.

Other Boundary Conditions

The only remaining points requiring difference formulas are the points at the edge of the flowfield. At the left boundary ϕ is arbitrarily set to 0. On the right boundary ϕ_n is specified to give the freestream condition above the cowl and inside the duct to give the correct compressor-face mass flow. On the axis, Eq. (3) is solved. Symmetry is used to generate the difference formula for ϕ_{rr}. At the upper surface it is assumed $\phi_r = 0$ (solid wall).

Line Relaxation

Line relaxation was chosen for two reasons; first, the most points can be solved simultaneously with a reasonable tradeoff on computer time and storage; and second, it eliminates some problems associated with variable mesh spacing. A question that arises is how to handle surface points with this scheme, and the best approach seems to be to solve those surface points adjacent to mesh points on the line, simultaneously with the line. A surface point by definition is adjacent to the next mesh point on the line that creates the surface point. Lines for the line relaxation run from one surface or the axis to another surface or the upper boundary.

To smooth the calculation for surface points some changes have been made in the way surface points are updated to handle surface points that are very close together. As an example (referring to Fig. 7) the value of ϕ for surface point S_1, ϕ_{S_1}, is calculated twice per sweep. It is first calculated when the line $m-1$ is solved, at which time if S_1 and S_2 are very close it will have a value approximately equal to ϕ_{S_2}. However, after the line m is updated there is a new value at S_2, $\phi_{s2}{}^{+}$ and $\phi_{s1}{}^{+}$ and $\phi_{s2}{}^{+}$ can be quite quite different although the points are extremely close together. To ensure that the values along the surface are smooth a new value is calculated for ϕ at S_1 using new values along the line m.

Initial Field and Starting

The calculation is started by initializing the field outside of the duct to uniform freestream flow. The flow inside the duct

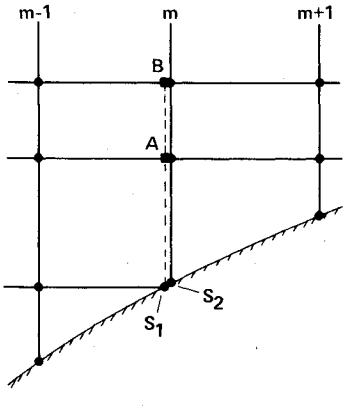

Fig. 7 Sequential updating of surface points.

is then initialized one-dimensionally to the correct compressor-face mass flow. The initial field as a result will in general have a discontinuity in the flowfield at the inlet entrance, and has flow through surfaces. The calculation is started at a very low Mach number and gradually brought to the correct Mach number by sweep 15. This allows the local flow to become aligned with adjacent surfaces before any of the upwind differencing is used. The calculation is run from the beginning with the same over-relaxation parameter.

Convergence Acceleration-Extrapolation

A technique described by Hafez and Cheng[12] is used to accelerate final convergence. A similar technique was also used by Caughey and Jameson.[8] It is based on theory for linear operators. The actual operator for this problem is nonlinear, but it behaves locally (in sweep number) like a linear operator. If k is the sweep counter, ϕ the matrix of unknowns, A a linear operator and b a constant matrix, the linear operator approximation of the equation is

$$\phi_{k+1}=A\phi_k+b \tag{29}$$

If x_i are the eigenvectors with associated eigenvalues λ_i ordered so $|\lambda_{i+1}|\leq|\lambda_i|$ the error $\epsilon_k=\phi-\phi_k$ can be written

$$\epsilon_k=\sum_i c_i x_i \tag{30}$$

With some manipulation it is found that

$$\epsilon_{k+1}=A\epsilon_k \tag{31}$$

and using the properties of eigenvectors

$$\epsilon_{k+K}=\sum_i \lambda_i{}^K c_i x_i \tag{32}$$

For large K if $|\lambda_1|>|\lambda_2|$

$$\epsilon_{k+K}\approx\lambda_1{}^K c_1 x_1 \tag{33}$$

and thus for $\bar{k}$ very large

$$\epsilon_{\bar{k}+1}\approx\lambda_1\epsilon_{\bar{k}} \tag{34}$$

if $C_k=\phi_k-\phi_{k-1}$ (the residue) the following can be shown

$$C_{\bar{k}+1}\approx(1-\lambda_1)\epsilon_{\bar{k}} \tag{35}$$

$$C_{\bar{k}+1}\approx\lambda_1 C_{\bar{k}} \tag{36}$$

If $\|\|$ is any norm then

$$\lambda_1\approx\frac{\|C_{k+1}\|}{\|C_k\|} \tag{37}$$

In practice, two norms are calculated for each sweep. One is the sum of the absolute values of the residues for all mesh points, and the other is the absolute value of the largest residue. When the eigenvalue λ, computed using Eq. (37), is steady for several sweeps for either of these norms and the eigenvalues for the two norms are nearly the same, the field is extrapolated using

$$\phi_p{}^{(\text{extrap})}=\phi_p{}^{(k-1)}+\frac{1}{1-\lambda}\left(\phi_p{}^{(k)}-\phi_p{}^{(k-1)}\right) \tag{38}$$

for all points p in the field.

This process has been very successful. It greatly decreases the number of sweeps required for convergence, and allows computation of solutions with much smaller residues than one could otherwise afford to calculate.

Sequenced Meshes

A considerable reduction in computer time is obtained by starting the relaxation process on a very coarse mesh. For the initial stages of the relaxation, the accuracy of the local solution is unimportant, because the solution is not close to the final solution. The solution is calculated on a coarse mesh until it begins to converge, and then the solution is interpolated to a finer mesh and continued until satisfactory convergence is obtained. This is generally when the program results are constant to three or four significant figures. The time savings is very large because, for a coarse mesh that has half the mesh lines of a finer mesh, the computing time to reach a given convergence level is about ⅛ that for the fine mesh. A factor of ¼ because there are ¼ as many points and a factor of ½ because the convergence per sweep is about twice as fast.

Computation Parameters

This analysis was programmed in FORTRAN IV for the CDC 6600 computer. The following parameters apply to a typical inlet with a freestream Mach number of 0.9 and a duct exit Mach number of 0.56. The final mesh was 62×39 and there are approximately 140 surface nodes. The number of computation nodes depends on the thickness of the cowl. Sweeps on the coarse mesh (32×20) take approximately 0.35 seconds and fine-mesh sweeps 1.13 seconds. The time does not go up by a factor of four because the number of surface and surface-adjacent points which require the most calculation increase by a little more than a factor of two for these geometries when going from the coarse to a fine mesh.

Most of the calculations made have taken 200 coarse-mesh sweeps, 150 fine-mesh sweeps and have involved several extrapolations of the field on each mesh. The solution was converged to three or four significant figures at this number of sweeps. The total run time using the RUN compiler is approximately four minutes and the program requires 71,000 octal words of core storage.

Results

The following examples indicate the program's capabilities. As stated previously, the program was intended as a step towards three-dimensional calculations and not as an end in itself. As a consequence a great amount of effort was spent in trying methods to optimize convergence, but very little work was spent in trying alternate meshes to improve accuracy or inserting extra mesh lines to better define shocks. This is because fast convergence is essential to a three-dimensional procedure. The addition of extra mesh or use of a finer mesh uses computer time and requires some extra coding, but does not involve any technical risk or help in the development of a three-dimensional program.

Program results were compared with the Boeing production program[7] which uses a streamline-coordinate transformation procedure, and results are comparable for the same meshes. It is expected better agreement with the data could be obtained by using finer meshes. Most of the test cases were run at a freestream Mach number of 0.9 because the higher-Mach-number flows are the most difficult to compute and thus the best test of the procedure.

Figures 8 and 9 show comparisons between the analysis and experiment for two Boeing inlet designs. The cowls were ¼-scale models (relative to a JT9D-3) and were tested in the Boeing 8 by 12 ft transonic wind tunnel.[13] Cowl 30 is a short (fan) duct with a thin lip. The contraction ratio is 9%. Cowl 3 is a ¾-length cowl with a thick lip (26% contraction ratio). The shocks are smeared over two mesh spacings and are in the correct position. The shocks can be made steeper by using more mesh lines in the vicinity of the shocks as the shock jump is always spread over approximately two mesh widths. The magnitude of errors is greatest for supersonic flow regions because the upwind differencing has much larger error terms than the subsonic central differencing. Use of a finer

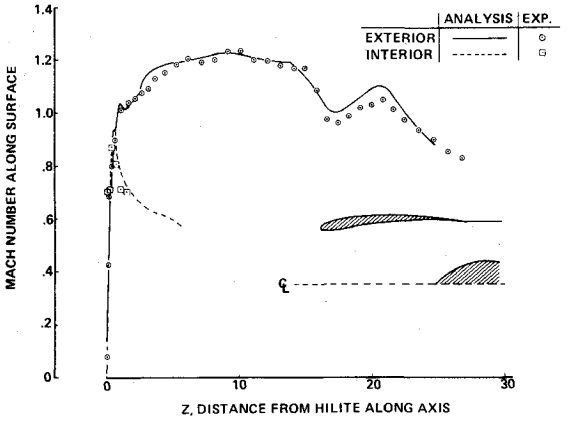

Fig. 8 Surface Mach number distribution, cowl 30, $M_\infty = 0.9$, $A_\infty / A_{HI} = 0.814$.

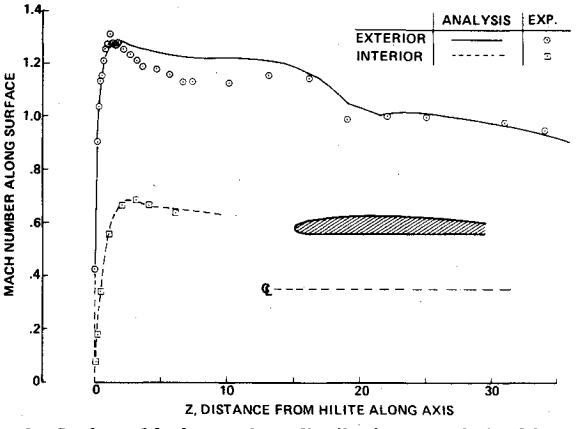

Fig. 9 Surface Mach number distribution, cowl 3, $M_\infty = 0.9$, $A_\infty / A_{HI} = 0.69$.

mesh in these regions would probably change the predictions slightly.

This analysis uses the nonconservative form of the potential flow equation. It has been shown[14,15] that the nonconservative form of the equation leads to under-prediction of the jump in flow properties across shocks, but this underprediction tends to be in the same direction as boundary-layer/shock-wave interaction effects so agreement with data is good. If one was to extend this analysis to include boundary-layer displacement-thickness effects one would have to carefully investigate to see if it would be better to use a conservative difference scheme, or as an alternative shock fitting.

Figure 10 shows results for one of the inlets reported by Arlinger.[6] Agreement with data is very good.

Figure 11 shows the comparison between the experiment and analysis for a 34% contraction ratio, 0.1738 scale, inlet model that was tested in the Boeing 9 by 9 ft Low Speed Wind Tunnel. This inlet and several others were evaluated for the CF6-50D engine on the YC-14 airplane. The particular results shown are for a static (zero forward speed) configuration for an airflow 9% greater than takeoff airflow. The different symbols for the data are for static taps at different circumferential locations. Agreement with the data is excellent. As with other results shown, the degree of agreement between data and analysis displayed requires careful location of mesh lines. This is a fault of the number of mesh lines used and not of the analysis in general. If more mesh lines were used the location of the mesh lines would not be as critical.

Conclusions

There are two sets of conclusions that have been made relative to this work. The first has to do with this work as it

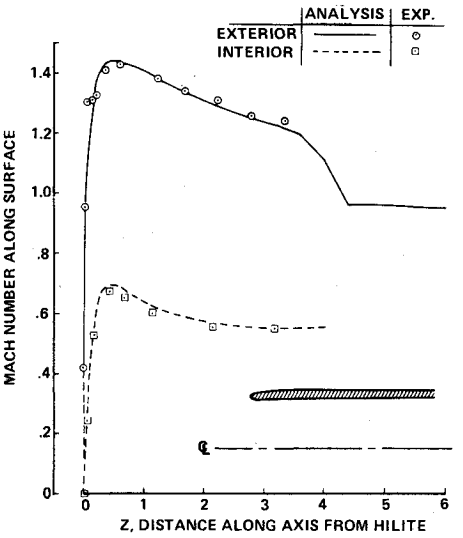

Fig. 10 Surface Mach number distribution, cowl L_{2-2}, $M_\infty = 0.9$, $A_\infty / A_{TH} = 0.847$.

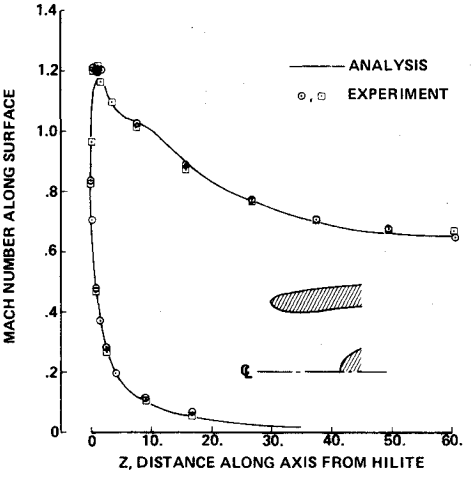

Fig. 11 Surface Mach number distribution, model 34, $M_\infty \approx 0$, $M_{TH} = 0.75$.

relates to predicting the flows analyzed, and the second is with regard to future work.

Present Work

This approach has been shown to be a valid one. Predicted results are in good agreement with experiment, and computer run times are very reasonable. As written, the program handles inlets with or without centerbodies with the cowl and centerbody of arbitrary shape. Modifications to handle two-dimensional flows or other axisymmetric bodies would be very easy. The problem of applying a Kutta condition has not been explored, but probably would not be too difficult. The program could easily handle an inlet with rings for example, if coding for a Kutta condition were available. The addition of more surfaces in the flowfield does not require any changes in the solution procedure.

Future Work

The program has been designed so that the method directly extends to three-dimensions. The most obvious problem connected with three-dimensions is the order-of-magnitude increase in computer time. It appears obvious that less points and a higher-order approach, possibly with shock fitting of some kind, would be desirable. With the present approach the use of more points allows a more accurate solution, but if the number of mesh lines is doubled the rate of convergence is

slowed by approximately a factor of two. For two-dimensional calculations with very fine meshes, and for three-dimensions for all but coarse meshes, one runs into the problem of having to use low-speed auxiliary storage devices (disk storage) to store part of the ϕ field, because there is space for only a fraction of the ϕ field in core storage. The use of out-of-core storage has an adverse affect on computer costs. For the three-dimensional case the additional computer coding required for a higher-order solution would require less storage space than a finer mesh would require. Since doubling the number of mesh lines leads to a factor of eight increase in the number of points and approximately a factor of two decrease in the convergence rate, the higher-order formulas could be roughly a factor of sixteen more complicated before the tradeoff on computation time becomes unfavorable.

Another possibility for improving the efficiency of the analysis would be to use a different mesh away from the body. The present approach uses a variable spaced grid which extends to the boundaries of the computational domain. If a very closely spaced mesh is used to maintain accuracy in a region near the body, this same close spacing also exists in the field far from the body because of the extension of the mesh lines. In the far field the close spacing is unnecessary and can be detrimental. The unnecessary points require time to compute and possibly slow convergence of the relaxation process. One possibility for improvement is that used by Boppe[16] where a fine mesh is used around the body (a wing in Boppe's work) and the far field is calculated with a coarse mesh. Another idea would be to terminate some mesh lines away from the body in order to have a more efficient mesh in the far field. The primary disadvantage of either approach is the increased complexity of the analysis.

Acknowledgment

The author wishes to acknowledge the assistance of R. G. Jorstad of Boeing Computer Services, Inc. who was responsible for the programming of the analysis.

References

[1]Murman, E. M. and Cole, J. D., "Calculation of Plane Steady Transonic Flows," *AIAA Journal*, Vol. 9, Jan. 1971, pp. 114-121.

[2]Magnus, R. and Yoshihara, H., "Inviscid Transonic Flow Over Airfoils," *AIAA Journal*, Vol. 8, Dec. 1970, pp. 2157-2162.

[3]Bailey, F. R. and Ballhaus, W. F., "Comparisons of Computed and Experimental Pressures for Transonic Flows About Isolated Wings and Wing-Fuselage Configurations," *Aerodynamic Analyses Requiring Advanced Computers, Part II*, NASA SP-347, 1975, pp. 1213-1231.

[4]Jameson, A., "Numerical Calculation of the Three Dimensional Transonic Flow Over a Yawed Wing," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, Palm Springs, Cal., 1973, pp. 18-26.

[5]Carlson, L. A., "Transonic Airfoil Analysis and Design Using Cartesian Coordinates," *Journal of Aircraft*, Vol. 13, May 1976, pp. 349-356.

[6]Arlinger, B. G., "Calculation of Transonic Flow Around Axisymmetric Inlets," *AIAA Journal*, Vol 13, Dec. 1975, pp. 1614-1621.

[7]Colehour, J. L., "Transonic Flow Analysis Using a Streamline Coordinate Transformation Procedure," AIAA Paper 73-657, Palm Springs, Cal., 1973.

[8]Caughey, D. A. and Jameson, A., "Accelerated Iterative Calculation of Transonic Nacelle Flowfields," AIAA Paper 76-100, Washington, D. C., Jan. 1976, submitted to *AIAA Journal*.

[9]South, J. C. and Jameson, A., "Relaxation Solutions for Inviscid Axisymmetric Transonic Flow Over Blunt or Pointed Bodies," *Proceedings of the AIAA Computational Fluid Dynamics Conference*, Palm Springs, Cal., July 1973, pp. 8-17.

[10]Lomax, H., and Steger, J. L., "Relaxation Methods in Fluid Mechanics," *Annual Review of Fluid Mechanics*, Vol. 7, Annual Reviews Inc., Palo Alto, Cal., 1975, pp. 63-88.

[11]Jameson, A., "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1," *Comm. on Pure and Applied Mathematics*, Vol. XXVII, 1974, pp. 283-309.

[12]Hafez, M. M., and Cheng, H. K., "Convergence Acceleration and Shock Fitting for Transonic Aerodynamics Computations," AIAA Paper 75-51, Pasadena, Cal., Jan. 1975; *AIAA Journal*, Vol. 15, March 1977, pp. 329-336.

[13]Madden, J., and Johnson, J. M., "High Speed Wing Tunnel Test Results for ¼-Scale Isolated Fan Cowls With ¾-Length Fan Ducts," The Boeing Co., Seattle, Wash., D6-22481, Aug. 1970.

[14]Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," *AIAA Journal*, Vol. 12, May 1974, pp. 626-633.

[15]Jameson, A., "Transonic Potential Flow Calculations Using Conservation Form," *Proceedings of the AIAA 2nd Computational Fluid Dynamics Conference,* Hartford, Conn., June 19-20, 1975, pp. 148-155.

[16]Boppe, C. W., "Calculation of Transonic Wing Flows by Grid Embedding," AIAA Paper 77-207, Los Angeles, Calif., 1977.