Computation of Transonic Potential Flow About Three-Dimensional Inlets, Ducts, and Bodies

Theodore A. Reyhner

MARCH 1982
Computation of Transonic Potential Flow About Three-Dimensional Inlets, Ducts, and Bodies

Theodore A. Reyhner
*Boeing Commercial Airplane Company*
*Seattle, Washington*

Prepared under Cooperative Agreement with NASA Langley Research Center and Boeing Commercial Airplane Company

NASA
National Aeronautics and Space Administration
Scientific and Technical Information Branch
1982
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>v</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 ANALYSIS</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Equation and Boundary Conditions</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Potential Flow Equation</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Boundary Conditions</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Weighting and Alternate Formulas</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Difference Quotients</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Formulas for First Derivatives (Velocities)</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Difference Quotients for $\phi_{xx}$, $\phi_{rr}$, and $\phi_{\theta\theta}$</td>
<td>11</td>
</tr>
<tr>
<td>2.3.3 Difference Quotients for $\phi_{xx}$, $\phi_{rr}$, and $\phi_{\theta\theta}$</td>
<td>14</td>
</tr>
<tr>
<td>2.3.4 Special $\delta$-Difference Quotients</td>
<td>17</td>
</tr>
<tr>
<td>2.4 Surface Derivatives</td>
<td>18</td>
</tr>
<tr>
<td>2.5 Field-Point Difference Equations</td>
<td>19</td>
</tr>
<tr>
<td>2.5.1 Axis</td>
<td>19</td>
</tr>
<tr>
<td>2.5.2 Modified Difference Equation for Field Points Adjacent to Surface</td>
<td>23</td>
</tr>
<tr>
<td>2.6 Surface-Point Difference Equations</td>
<td>24</td>
</tr>
<tr>
<td>2.7 Solution Procedure</td>
<td>26</td>
</tr>
<tr>
<td>2.7.1 Code Organization</td>
<td>27</td>
</tr>
<tr>
<td>2.7.2 Procedure for Sweeping</td>
<td>28</td>
</tr>
<tr>
<td>2.7.3 Convergence Acceleration</td>
<td>28</td>
</tr>
<tr>
<td>2.8 Geometry Specification</td>
<td>30</td>
</tr>
<tr>
<td>2.9 Results</td>
<td>30</td>
</tr>
<tr>
<td>2.10 Accuracy</td>
<td>35</td>
</tr>
<tr>
<td>2.11 Conclusions</td>
<td>42</td>
</tr>
<tr>
<td>3.0 USE OF THE PROGRAM</td>
<td>43</td>
</tr>
<tr>
<td>3.1 Computational Parameters</td>
<td>43</td>
</tr>
<tr>
<td>3.2 Input Format</td>
<td>43</td>
</tr>
<tr>
<td>3.2.1 Input-Group Summary</td>
<td>44</td>
</tr>
<tr>
<td>3.2.2 Input-Group Descriptions</td>
<td>45</td>
</tr>
<tr>
<td>3.2.3 Geometry Specification</td>
<td>51</td>
</tr>
<tr>
<td>3.2.4 Mesh Specification</td>
<td>51</td>
</tr>
<tr>
<td>3.2.5 Geometry Limitations</td>
<td>54</td>
</tr>
<tr>
<td>3.2.6 Sample Input Files</td>
<td>57</td>
</tr>
<tr>
<td>3.3 Program Output</td>
<td>57</td>
</tr>
<tr>
<td>3.4 Run Quality</td>
<td>74</td>
</tr>
<tr>
<td>3.5 Diagnostics and Troubleshooting</td>
<td>75</td>
</tr>
</tbody>
</table>
SUMMARY

An analysis has been developed and a computer code, P465 Version A, written for predicting three-dimensional transonic potential flow about inlets, ducts, and bodies. The basic approach is very general and essentially independent of geometry. The analysis, as programmed, uses cylindrical coordinates and admits boundary conditions for the aforementioned configurations. Solutions of the full potential equation are obtained using finite differences and successive line over-relaxation (SLOR). Extrapolation and a sequence of increasingly denser grids are used to accelerate convergence and thus decrease the computer time required to obtain a solution.

The analysis has been programmed in extended FORTRAN IV for the Control Data Corporation CYBER 203 computer. Use of reasonably dense meshes for three-dimensional computations requires either a very large computer, such as the CYBER 203, or a great amount of complex software to use disk memory for backup of core memory. The run time on the CYBER 203 for a typical inlet calculation is about five minutes. Some special features of the CYBER 203 have been used in the computer code to maximize efficiency of the code. Use of the analysis on another super computer, such as the CRAY 1, is practical but would require a certain amount of recoding.

Comparisons between computer results and experimental measurements are presented to verify the capabilities of the code. Agreement between analysis and experiment is excellent for flowfields which are essentially inviscid and irrotational. The greatest disagreements between the analysis and experiment occur for flows where viscous (boundary-layer) effects are significant.

Program input and output formats are described and examples are presented.

This report does not discuss geometry specification. The analysis code requires input of a computational mesh and all intersections of the computational mesh with surfaces, including coordinates and components of the surface normal. This geometry specification can come from any source and the file of intersections can be in any order. The program does extensive checking of the geometrical input.
## NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>speed of sound</td>
</tr>
<tr>
<td>$C_p$</td>
<td>coefficient of pressure, $(p/p_\infty - 1)^{\gamma/\gamma M^2}$</td>
</tr>
<tr>
<td>M</td>
<td>Mach number</td>
</tr>
<tr>
<td>$M_{C.F.}$</td>
<td>Mach number at compressor face for uniform flow parallel to axis</td>
</tr>
<tr>
<td>$\vec{n}$</td>
<td>unit normal to surface oriented into flowfield</td>
</tr>
<tr>
<td>$n_x,n_r,n_\theta$</td>
<td>components of $\vec{n}$</td>
</tr>
<tr>
<td>p</td>
<td>static pressure</td>
</tr>
<tr>
<td>q</td>
<td>velocity, $(u^2 + u_r^2 + u_\theta^2)$</td>
</tr>
<tr>
<td>r</td>
<td>radius</td>
</tr>
<tr>
<td>$r_{FAN}$</td>
<td>radius of engine fan</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>$r$ at outer edge of computational cylinder</td>
</tr>
<tr>
<td>s</td>
<td>arc length</td>
</tr>
<tr>
<td>u</td>
<td>axial velocity component, $u = \phi_x$</td>
</tr>
<tr>
<td>$u_r$</td>
<td>radial velocity component, $u_r = \phi_r$</td>
</tr>
<tr>
<td>$u_\theta$</td>
<td>component of velocity along cut of surface</td>
</tr>
<tr>
<td>$u_\theta$</td>
<td>circumferential velocity component, $u_\theta = \frac{1}{r} \phi_\theta$</td>
</tr>
<tr>
<td>v</td>
<td>velocity component in y direction, $v = \phi_y$</td>
</tr>
<tr>
<td>w</td>
<td>velocity component in z direction, $w = \phi_z$</td>
</tr>
<tr>
<td>x</td>
<td>axial coordinate</td>
</tr>
<tr>
<td>y</td>
<td>coordinate, $y = r \cos \theta$</td>
</tr>
<tr>
<td>z</td>
<td>coordinate, $z = r \sin \theta$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>angle of attack, $v_x/u_\infty$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>angle of yaw, $w_z/u_\infty$</td>
</tr>
<tr>
<td>$\beta, \beta_x, \beta_1, \ldots$</td>
<td>weighting parameters</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>ratio of specific heats</td>
</tr>
</tbody>
</table>
$\Delta x, \Delta r,...$ step size in $x, r$ etc

$\Delta \phi$ change in $\phi$ between relaxation sweeps

$\eta$ nondimensional coordinate, $0 \leq \eta \leq 1.0$

$\theta$ circumferential coordinate

$\theta_\xi$ value of $\theta$ for additional plane of symmetry

$\xi$ radial coordinate along plane of symmetry, $\theta = \theta_\xi$

$\lambda$ extrapolation factor

$\phi$ potential function

$\phi_n$ velocity normal to surface

$\phi_s$ $\partial \phi / \partial s$, velocity in direction of $s$

$\phi_{s_x}, \phi_{s_r}, \phi_{s_\theta}$ velocity components for constant $x, r, \text{and } \theta$ cuts of surface respectively

$\omega_x$ local $x$ over-relaxation parameter

$\overline{\omega}_x$ reference $x$ over-relaxation parameters ($\overline{\omega}_x = 1.85$)

$\omega_\theta$ local $\theta$ under-relaxation parameter

$\overline{\omega}_\theta$ reference $\theta$ under-relaxation parameter ($\overline{\omega}_\theta = 0.9$)

**Superscript**

+ value after updating

**Subscripts**

$i$ index for $x$ mesh values

$j$ index for $r$ mesh values

$k$ index for $\theta$ mesh values

$l,l',u$ step-size subscripts

$A,B,P,P'...$ points on grid or surface

$S,S_1,S_2$ surface points

$x,y,z,r,\theta,s$ partial derivatives

$0,1,2$ step-size subscripts
\[ \infty \quad \text{freestream} \]

**Special**

\[ \phi_{x|S} \quad \phi_x \text{ at point } S \]

\[ \mathcal{Q} \quad \text{centerline (axis)} \]

\[ \| \quad \| \quad \text{norm} \]
LIST OF FIGURES

1. Typical Coarse Computational Mesh for a Constant $\theta$ Cut in the Vicinity of the Inlet ................................................................. 3
2. Geometry and Boundary Conditions for Inlet Flowfield Computation .......................... 7
3. Nomenclature for Typical Difference Quotients .............................................. 10
4. Example Configurations for Upwind Differentiating of Cross-Derivatives .................. 16
5. Notation for the Axis ................................................................................... 22
6. Nomenclature for Typical Surface-Point Boundary-Condition Equation .................... 25
7. Graphical Display of V/STOL Airplane Inlet Geometry ...................................... 31
8. Cowl Surface Mach Number Distribution for an Asymmetric V/STOL Airplane Inlet 32
9. Cowl Surface Mach Number Distribution for an Asymmetric V/STOL Airplane Inlet 33
10. Cowl Surface Mach Number Distribution for an Asymmetric V/STOL Airplane Inlet 34
11. Computational Flowfield and Mesh in the Vicinity of a Commercial-Transport Turbofan-Engine Type Inlet ................................................................. 36
12. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet .............................................. 37
13. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet .............................................. 38
14. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet .............................................. 39
15. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet .............................................. 40
16. View of Upper Surface of a Mixer Lobe ....................................................... 41
17. Standard Coarse Mesh for an Inlet Flowfield Calculation ...................................... 53
18. Mesh-Density Limitations Between Surfaces
   a) Mesh Too Coarse, Code Will Fail ............................................................. 55
   b) Mesh Minimal, but Code Will Work ......................................................... 55
19. Mesh-Density Limitations Relative to Surface Geometry
   a) Inadequate Mesh for Geometry Resolution ............................................... 56
   b) Adequate Mesh for Geometry Resolution .................................................. 56
20. Sample Input Files
   a) Truncated Input File For An Inlet ............................................................ 58
   b) Truncated Input File For A Mixer Lobe ...................................................... 59
21. Sample Output
   a) Sample Output – Introductory Section ...................................................... 60
   b) Convergence History .............................................................................. 64
   c) Timing Information ................................................................................. 65
   d) Surface Properties – $\theta$-Constant Cuts .................................................. 66
   e) Field Properties ....................................................................................... 67
   f) Mass-Flow Conservation ......................................................................... 68
   g) Surface Properties – All Cuts ................................................................. 69
22. Typical Geometry Problems
   a) Missing Surface Point ................................................................................ 76
   b) Inconsistent Surface Points ...................................................................... 76
   B-1 Rectangular Box for Interpolation .......................................................... 82
LIST OF TABLES

1. Terms and Difference Quotients for Field-Point Difference Equations ....... 20
2. Terms for the Difference Equation in Cartesian Coordinates ................ 21
3. Standard Coarse Mesh for an Inlet ........................................... 52
4. Headings for Surface Point and Surface Geometry Printout ................... 71
5. Convergence History Headings ............................................ 72
6. Surface Point Printout Headings ........................................... 73
7. Field-Point Printout Headings ............................................. 74
1.0 INTRODUCTION

This report describes a computer code, P465 Version A, for predicting the inviscid irrotational flowfield about a three-dimensional inlet, duct, or body. The flowfield is generated by a finite-difference, line-relaxation solution of the complete potential flow equation. The predicted results are excellent approximations of actual flowfields where viscous effects are small, i.e., flows with thin attached boundary layers.

During the past decade there has been tremendous progress in the field of transonic potential-flow computation. The work started with Murman and Cole\(^1\) who used upwind differencing in supersonic flow regions to solve mixed elliptic and hyperbolic equations. Their work was based on small disturbance theory where the flow was principally aligned with one coordinate. Jameson\(^2\) extended this approach to the full potential equation with arbitrary flow direction. Several classes of three-dimensional geometries have been solved using the full potential equation\(^2\)\(^-\)\(^5\). Most extensions to three dimensions have been based on some mapping from the physical coordinates to a more convenient computational coordinate system. If such mappings can be found, the computational problem, exclusive of the generation of the mapping, can be greatly simplified. This approach becomes more difficult as the geometry of objects to be considered becomes more complex, because it becomes more difficult to find convenient mappings that will provide reasonable coordinates for the computation. A further disadvantage with a complex mapping is that it can be very difficult to have any physical understanding of the solution process.

The approach used for this analysis separates the coordinate system and the geometry. Thus, this approach does not require the finding of a new transformation for every family of geometries. The method was originally tested for axisymmetric problems\(^6\) and later applied to axisymmetric inlets at angle of attack\(^3\). This report describes the extension of the method to general three-dimensional geometries. The method also has the advantage that the solution is obtained in a physical coordinate system with physical variables and hence the entire solution process is more understandable. The analysis, as programmed, uses cylindrical coordinates since they are suited to nacelle calculations, but the analysis could easily be changed to use a Cartesian, skewed Cartesian, or any other coordinate system.

The analysis is independent of the coordinate system except that if the coordinate system is a poor choice for the problem, accuracy will suffer as there will be insufficient mesh nodes. The practical limit on the number of mesh nodes is dictated by computational time. The nature of the relaxation process imposes an exponential increase in computational time with increasing mesh density. This severely limits the number of mesh nodes and makes it essential that the coordinate system be such that the mesh nodes are well placed. The cylindrical coordinate system used in the computer code gives reasonable node placement for nacelle, duct, and body geometries where such geometries are approximately bodies of revolution. Cylindrical coordinates can be used for geometries that are not nearly bodies of revolution, but they may not be very efficient.
2.0 ANALYSIS

Solutions to the full partial differential equation for compressible potential flow along with appropriate boundary conditions are desired for three-dimensional flow. Approximate solutions of this equation are obtained by replacing the partial differential equation and boundary conditions with finite-difference equations for the value of the potential function, at discrete points in the flowfield and on the boundary. The finite-difference equations are obtained by replacing partial derivatives in the differential equations with difference quotients. The difference equations are coupled, nonlinear, algebraic equations in the value of the potential function at field and surface mesh points. The nonlinear difference equations are solved by an iterative process to the desired accuracy. The degree of approximation of the difference solution to the true solution of the partial differential equation is a function of the mesh spacing.

The solution technique can use an arbitrary mesh as long as the mesh is dense enough to ensure that there are several mesh nodes between independent surfaces. As programmed, a cylindrical mesh with variable mesh-line spacing is used because it is relatively efficient for inlet and nacelle configurations. A typical constant θ cut showing the mesh and an inlet cowl is displayed in figure 1. The mesh is not required to intersect with the body in any predescribed manner. Nodes used in the calculation are mesh intersects in the flowfield and intersections of mesh lines with surfaces. Difference quotients are written to take into account variable spacing between field and surface nodes.

The difference equations consist of a large number of coupled, nonlinear, algebraic equations. The direct solution of such a set of equations is certainly impractical if not impossible. A multistep process is used to develop a solution procedure for the difference equations. The basic approach uses an iterative technique, starting with an initial guess, computing an approximate solution, and then repeating the cycle or sweep with the approximate solution used to compute a better approximate solution. As a minimum requirement, the iteration technique has to be convergent. In practice, for the three-dimensional transonic potential-flow problem, the iteration technique has to converge rapidly enough that the computation time is practical.

The first step in the iterative solution process is to linearize the difference equations. This is done by dividing terms into coefficients and unknowns, and using the initial guess or solution for the previous iteration to approximate the coefficients. The linear equation system that results is fully coupled, or implicit, and would have to be solved simultaneously. Simultaneous solution of such a large system of linear equations is not practical. As a consequence, the equations have to be at least partly decoupled by substituting old values of the potential function, φ, for some of the (unknown) values of φ (denoted by φ⁺). How this substitution is done affects stability, convergence, convergence rate, and the practicality of the solution procedure. The process of mixing old and new values and iterating (sweeping) to obtain a solution is commonly called relaxation and is discussed in many references. (See ref. 7, for example.) Generally, the more thoroughly the equations are decoupled, the less work is required to take one relaxation sweep; however, the more relaxation sweeps are required to reach a given level of convergence or degree of accuracy of the solution. A fully decoupled system, where only one equation needs to be solved at a time, is known as explicit and generally exhibits very slow convergence properties.
Figure 1. Typical Coarse Computational Mesh for a Constant $\theta$ Cut in the Vicinity of the Inlet
A decoupling of the equation system so that all the equations along one mesh line are solved simultaneously is known as line relaxation. It is a popular technique because there exists a very efficient algorithm known as the Thomas or tridiagonal algorithm to obtain the solution for such an equation system. The relaxation process can be accelerated or stabilized, as desired or required, by processes known as over- or under-relaxation, respectively. Over-relaxation consists of taking the difference of the new and previous solutions, multiplying the difference by a factor between one and two, and adding the difference to the previous solution to create an improved new solution. Under-relaxation is similar except the multiplication factor is less than one. In this code over-relaxation is used for one term of the partial differential equation and under-relaxation for another. The combination of line relaxation and over-relaxation is commonly called successive line over-relaxation (SLOR). This is basically the technique used in the analysis discussed herein.

A penalty involved in using a line-relaxation scheme on a vector computer is that the tridiagonal algorithm, used to solve the simultaneous equations for a line, does not vectorize efficiently. However, the ratio of computing time spent solving the equations with the tridiagonal solver, versus the time spent calculating the equation coefficients, is small. Also, the SLOR scheme takes many less relaxation sweeps to reach a given level of convergence than an explicit technique. A sweep is a complete solution in sequential order of all the difference equations for the flowfield.

Each relaxation sweep is computed using SLOR. Additional techniques have also been employed to accelerate convergence. One such technique is extrapolation using intermediate solutions. Extrapolation is based on the theory of linear algebraic equations and the existence of eigenvalues and eigenvectors. When the solution process appears to be developing in a particular manner it can be extrapolated to obtain a much better approximation of the final solution. Another technique is the use of a sequence of meshes of increasing density. The initial field is assumed to be uniform flow and is applied on a very coarse mesh. Relaxation is continued on this mesh until partial convergence is obtained. The initial mesh is too coarse to be used to compute an accurate solution to the flow problem, but the solution from the coarse mesh is a much better starting guess for the next mesh than uniform flow. The advantage of using a sequence of meshes is that computation is cheaper per sweep, and convergence is faster on a coarser mesh.

Along with the consideration of the practicality of making a relaxation sweep, the linearization and decoupling of the difference equations has to be such that the relaxation process converges. One aspect critical to the transonic flow solution is the use of upwind differencing where the flow is supersonic.

The organization of this analysis, as programmed, is biased by the strengths and weaknesses of the computer for which the code was originally written. In particular, the CDC CYBER 200 family of computers has large memory and operates most efficiently with long-vector operations. The large memories make three-dimensional codes possible. The efficiency of long-vector operations makes it desirable to decouple the equations so that many coefficients can be calculated at once.
2.1 EQUATION AND BOUNDARY CONDITIONS

The equation to be solved is the complete equation for inviscid, irrotational flow formulated in terms of a velocity potential, $\phi$. This form of the equation has the significant advantage that there is only one unknown that has to be stored at mesh nodes. This is particularly important for a three-dimensional elliptic problem where computer storage is very critical.

2.1.1 POTENTIAL FLOW EQUATION

The equation for the velocity potential in cylindrical coordinates is

\[
\left( a^2 - \phi_x^2 \right) \phi_{xx} + \left( a^2 - \phi_r^2 \right) \phi_{rr} + \left( a^2 - \frac{\phi_\theta^2}{r^2} \right) \frac{\phi_{\theta \theta}}{r^2} - 2 \phi_x \phi_r \phi_{xr} \\
- 2 \phi_r \phi_{rr} - 2 \phi_\theta \phi_{\theta x} + \left( a^2 + \frac{\phi_r^2}{r^2} \right) \frac{\phi_r}{r} = 0
\]

(1)

where $\phi$ is the velocity potential and the local speed of sound, $a$, is given by

\[
a^2 = a_\infty^2 - \frac{\gamma - 1}{2} \left( \phi_x^2 + \frac{\phi_r^2}{r^2} - \phi_\theta^2 \right).
\]

(2)

The potential equation in Cartesian coordinates is

\[
\left( a^2 - \phi_x^2 \right) \phi_{xx} + \left( a^2 - \phi_y^2 \right) \phi_{yy} + \left( a^2 - \phi_z^2 \right) \phi_{zz} - 2 \phi_x \phi_y \phi_{xy} \\
- 2 \phi_y \phi_z \phi_{yz} - 2 \phi_x \phi_z \phi_{xz} = 0.
\]

(3)

This form of the equation is used in generating difference equations for points on the axis.

The velocity components in the flowfield $(u, u_r, u_\theta)$ for the cylindrical coordinate system are obtained from the potential function with the following relationships:

\[
u = \phi_x,
\]

\[
u_r = \phi_r,
\]

\[
u_\theta = \frac{\phi_\theta}{r}.
\]

(4)
The velocity components for a Cartesian reference frame \((u, v, w)\) are obtained using the equations,

\[
\begin{align*}
v &= \phi_y = u_r \cos \theta - u_\theta \sin \theta \\
w &= \phi_z = u_r \sin \theta + u_\theta \cos \theta.
\end{align*}
\] (5)

2.1.2 BOUNDARY CONDITIONS

The boundary condition at solid surfaces is that the velocity normal to the surface, \(\phi_n\), equals zero. The boundary condition at the exit of a duct, or at the compressor face of an inlet, is that the axial velocity is fixed at the uniform value that gives a specified mass flow. At the left boundary of the computational field, the potential function, \(\phi\), is specified. For an inlet flowfield computation (figure 2), at the left boundary,

\[
\phi = u_\infty x + v_\infty r \cos \theta + w_\infty r \sin \theta + \text{constant.}
\] (6)

This is also the equation for the initial flowfield approximation.

For an isolated inlet in a freestream, the following additional boundary conditions are used. Equation 6 is used to specify \(\phi\) for 90 degrees \(\leq \theta \leq 270\) degrees and \(r = r_{\text{max}}\). The outflow velocity, \(\phi_n\), for the computational cylinder is specified by

\[
\phi_n \big|_{r = r_{\text{max}}} = \phi_r \big|_{r = r_{\text{max}}} = v_\infty \cos \theta + w_\infty \sin \theta
\] (7)

for 0 degrees \(\leq \theta < 90\) degrees and 270 degrees \(< \theta \leq 360\) degrees and \(r = r_{\text{max}}\). \(\phi_x\) is specified as equal to \(u_x\) at the right boundary of the computational field outside of the inlet.

The boundary conditions on the far field for an isolated nacelle or body are only approximate. The correct boundary condition is that the velocities approach the freestream values far from the body. This boundary condition cannot be directly satisfied for a potential-flow solution in a finite computational volume. As written, the boundary conditions specify either the normal or tangential component of velocity at any surface. In most cases the normal component of velocity is specified for outflow surfaces and the tangential component for inflow surfaces. The exceptions are nonzero yaw angle or negative angle of attack. For these cases, the boundary conditions on some of the surfaces could be changed; for example, the analysis is convergent for a negative angle of attack which essentially reverses the boundary conditions on the top and bottom of the computational cylinder. Changes to the far-field boundary conditions would probably affect accuracy and convergence rate of the solution, however, no extensive studies of these effects have been made.
\[ \phi_r = v_\infty \cos \theta + w_\infty \sin \theta \]

\[ r = r_{\text{max}} \quad 0^\circ \leq \theta < 90^\circ \]

AND \( 270^\circ < \theta \leq 360^\circ \)

\[ \phi_x = u_\infty \]

\[ \phi_n = 0 \]

\[ \phi \text{ SPECIFIED} \quad r = r_{\text{max}} \quad 90^\circ \leq \theta \leq 270^\circ \]

\[ \text{COMPRESSOR FACE} \]

\[ \phi_x \text{ SPECIFIED} \]

Figure 2. Geometry and Boundary Conditions for Inlet Flowfield Computation
The present far-field boundary conditions are satisfactory provided the boundaries are located far enough from the body. Locating the far-field boundaries five to ten nacelle diameters away from the body is usually sufficient to ensure negligible effects of the boundary conditions on the computed flowfield near the body. If the far-field boundary is so placed, the effects of the boundary conditions are of the order of or smaller than other approximations such as treating a nacelle as an isolated nacelle.

2.2 WEIGHTING AND ALTERNATE FORMULAS

A possible problem with this approach to solving the potential flow equation is that the step size between a surface point and an adjacent field point is uncontrolled and can become arbitrarily small. Derivatives are calculated by dividing by the step size and thus any error in the potential function, $\phi$, can be magnified by the reciprocal of the step size, which can become arbitrarily large. This can cause accuracy and stability problems if not considered. To avoid this problem, alternate formulas that do not use the field point $\phi$ value are used when the step size to a surface point is very small. To obtain a smooth solution process, a weighted combination of the regular and alternate formulas is used if the surface points are closer to the field points than one-half the local field-point spacing. This procedure, which is called weighting in this document, has been used for many difference quotients. Typical formulas are presented in this report.

The $\phi$ values at field points adjacent to surface points are used in differencing at other field nodes, even when they are very close to surface nodes. It is essential that as a field and surface node become close, the $\phi$ values at these nodes approach the same value. This behavior is ensured by calculating the $\phi$ value at the field node by interpolation between the surface value and other field-point values, if the spacing to the surface is very small. A weighted combination of interpolation and the usual field-point difference equation is used when the spacing between the field node and any surface node is less than one-half the local mesh spacing.

2.3 DIFFERENCE QUOTIENTS

Difference equations are generated by replacing partial derivatives in the partial differential equation with difference quotients. The choice of difference quotients is not arbitrary, nor is it unique. The difference quotients chosen are based on several, often conflicting, criteria. First, the difference quotients chosen must lead to a stable convergent solution process. Second, the use of more nodes in a difference quotient can give greater accuracy, but normally at the price of greatly increased complexity. For this analysis, where the body can intersect the mesh arbitrarily, the use of complex, many-point difference quotients would be extremely difficult.

The highest-order derivatives in the partial differential equation are second derivatives, which require at least three points in the difference quotients. As any more than three points in the difference quotients would greatly complicate many aspects of the problem, three-point difference quotients have been used.

There are three general categories of difference quotients: central, forward, and backward differences. Categorization is based on where the additional nodes used in the differencing lie with respect to the node at which the derivative is to be evaluated. Forward or backward differences relate to the use of additional nodes which lie ahead of or behind, in time or space, the node where the derivative is to be evaluated. In general, central differencing is preferred because of smaller truncation errors, however, in some
situations, central-difference quotients may result in unstable difference equations. In particular, for this problem, stability requires the use of one-sided differences for second derivatives where the flow is supersonic. These differences are termed "upwind" as the direction of the differencing is opposite the flow direction.

2.3.1 FORMULAS FOR FIRST DERIVATIVES (VELOCITIES)

The difference quotients for first-degree partial derivatives are central differenced for greater accuracy. Three-point differences are used giving second-order accuracy with truncation errors proportional to the square and/or product of the two step sizes involved. If surface points are used, alternate formulas may be weighted into the calculation to avoid problems due to dividing by the step size to the surface when the step size becomes very small. The first derivatives represent the flow velocities except for \( \phi_\theta \) which equals \( ru_\theta \). Difference quotients and weighting formulas are given here for \( x \) derivatives. The \( r \) and \( \theta \) derivatives are calculated the same way. Typical configurations for differencing are shown in figure 3.

All old \( \phi \) values are used in the first-derivative calculations, except for the calculation of \( \phi_x \) in the last term of equation 1. Old \( \phi \) values are used for the first derivatives because they are used to calculate the coefficients of the terms in the difference equations before the difference equations are solved.

For a regular field point (or a field point that is not adjacent to a surface point in the \( x \) coordinate) (fig. 3a)

\[
\phi'_x \bigg|_{i,j,k} = \frac{\Delta x^2 \phi'_{i+1,j,k} + (\Delta x^2 - \Delta x^2) \phi'_{i,j,k} - \Delta x^2 \phi'_{i-1,j,k}}{\Delta x \Delta x (\Delta x + \Delta x)}.
\] (8)

For a field point that is surface-adjacent in \( x \) (fig. 3b)

\[
\phi'_x \bigg|_A = \frac{\Delta x^2 \phi'_S + (\Delta x^2 - \Delta x^2) \phi'_A - \Delta x^2 \phi'_B}{\Delta x_1 \Delta x_2 (\Delta x_1 + \Delta x_2)}.
\] (9)

\[
\phi'_x \bigg|_A = \frac{1}{\Delta x_1 + \Delta x_2} \left[ 2 \Delta x_1 \frac{\phi'_S - \phi'_B}{\Delta x_1 + \Delta x_2} + (\Delta x_2 - \Delta x_1) \phi'_x \bigg|_S \right].
\] (10)

and

\[
\phi'_x \bigg|_A = \beta \phi'_x \bigg|_A + (1-\beta) \phi'_x \bigg|_A
\] (11)
Figure 3. Nomenclature for Typical Difference Quotients
where

\[
\beta = \begin{cases} 
\frac{\Delta x_1}{2\Delta x_0} & \Delta x_1 < \frac{1}{2}\Delta x_0 \\
1.0 & \Delta x_1 \geq \frac{1}{2}\Delta x_0 
\end{cases}
\]

\(\psi_{x|S}\) is the value of \(\psi_x\) for the surface point \(S\) that is calculated from the surface-point \(\phi\) values according to the procedure described in section 2.4.

If the field point is surface-adjacent in \(x\) in the opposite direction as well (fig. 3c)

\[
\phi_x^{(3)}|_A = \frac{\frac{\Delta x_1 \phi_x}{S'} + \frac{\Delta x_2 \phi_x}{S}}{\Delta x_1 + \Delta x_2}
\]

\[
\beta' = \begin{cases} 
\frac{\Delta x_2}{\frac{1}{2}\Delta x_0'} & \Delta x_2 < \frac{1}{2}\Delta x_0' \\
1.0 & \Delta x_2 \geq \frac{1}{2}\Delta x_0'
\end{cases}
\]

\[
\beta'' = \text{minimum}(\beta, \beta')
\]

and

\[
\phi_x|_A = \beta'' \phi_x^{(1)}|_A + (1-\beta'') \phi_x^{(3)}|_A .
\]

2.3.2 DIFFERENCE QUOTIENTS FOR \(\phi_{xx}, \phi_{rr}, \text{AND} \phi_{\theta\theta}\)

The difference quotients used for replacing the second derivatives affect the stability of the relaxation scheme. Central differencing is preferred for accuracy, but leads to unstable behavior if the flow becomes supersonic. The technique for avoiding this problem is to switch to upwind difference quotients when the flow becomes supersonic.

It would be desirable if all new \(\phi\) values could be used in the difference quotients, but that leads to a coupled equation set that is impractical to solve. The equations presented here are coupled only along radial lines and are of tridiagonal form. There is a relatively simple, well-known, efficient, algorithm for performing a direct solution of tridiagonal systems. The algorithm is presented in appendix A. For \(x\) derivatives, new values are used at \(i-1\) because they are available and an over-relaxation parameter, \(\omega_x\), is used to speed convergence for the subsonic central-difference quotient. The central-difference quotient for \(\phi_{\theta\theta}\) uses old values at \(k-1\) and \(k+1\) in order that all coefficients for an \(x\) equals constant plane can be calculated at one time. This is desirable to exploit the vector speed of the CYBER 203. An under-relaxation parameter, \(\omega_{\theta}\), is used to maintain stability.
Central difference formulas:

\[
\phi_{xx} = \left\{ \Delta x \phi_{i+1,j,k}^+ - (\Delta x + \Delta u) \left[ \frac{1}{\omega_x} \phi_{i,j,k}^+ \right] \right. \\
+ \left. \left( 1 - \frac{1}{\omega_x} \right) \phi_{i,j,k} \right\} + \Delta x \phi_{i-1,j,k}^+ \\
\div \left[ \frac{1}{2} \Delta x \Delta u (\Delta x + \Delta u) \right] \\
\omega_x = 1 + \left( \frac{\Delta x + \Delta u}{\max (\Delta x, \Delta u)} - 1 \right) (\bar{\omega}_x - 1) \tag{15}
\]

\[
\phi_{xx} = \left\{ \Delta x \phi_{i,j+1,k}^+ - (\Delta x + \Delta u) \left[ \frac{1}{\omega_x} \phi_{i,j,k}^+ \right] \right. \\
+ \left. \left( 1 - \frac{1}{\omega_x} \right) \phi_{i,j,k} \right\} + \Delta x \phi_{i,j-1,k}^+ \\
\div \left[ \frac{1}{2} \Delta x \Delta u (\Delta x + \Delta u) \right] \\
\omega_x = 1 + \left( \frac{\Delta x + \Delta u}{\max (\Delta x, \Delta u)} - 1 \right) (\bar{\omega}_x - 1) \tag{16}
\]

\[
\phi_{\theta \theta} = \left\{ \Delta \theta \phi_{i,j,k+1}^+ - (\Delta \theta + \Delta \theta) \left[ \frac{1}{\omega_\theta} \phi_{i,j,k}^+ \right] \right. \\
+ \left. \left( 1 - \frac{1}{\omega_\theta} \right) \phi_{i,j,k} \right\} + \Delta \theta \phi_{i,j,k-1}^+ \\
\div \left[ \frac{1}{2} \Delta \theta \Delta \theta u (\Delta \theta + \Delta \theta u) \right] \\
\omega_\theta = 1 + \left( \frac{\Delta \theta + \Delta \theta u}{\max (\Delta \theta, \Delta \theta u)} - 1 \right) (\bar{\omega}_\theta - 1) \tag{17}
\]

\(\bar{\omega}_x\) and \(\bar{\omega}_\theta\) are reference over- and under-relaxation parameters. They are the values used for a uniform mesh. The values \(\omega_x\) and \(\omega_\theta\) have been modified for the local mesh spacing.
The same formulas are used if a surface point, or points, are involved in the differencing, except that the appropriate surface point \( \psi \) value(s) and step size(s) are substituted.

The upwind difference formulas for \( \phi_{xx} \), \( \phi_{rr} \), and \( \phi_{\theta\theta} \) use a mixture of old and new \( \phi \) values for stability and improved rate of convergence. Where a new value of \( \phi \) is shown in a formula, an old value is used, if the flow direction is such that a new value is not available.

The switch between central and upwind differencing takes place as the coefficient of the second derivative changes sign. This leads to a smooth stable behavior of the solution process. As an example, central differencing is used for \( \phi_{xx} \) if \( |\phi_x| < a \) and upwind differencing if \( |\phi_x| > a \).

Upwind difference formulas are shown only for \( \phi_{xx} \) for one flow direction. Difference quotients for \( r \) and \( \theta \) and other flow directions are similar. For a point that is regular in \( x \) and for \( |\phi_x| > a \),

\[
\phi_{xx} = \frac{(\Delta x_2 + \Delta x_2)(\phi_{i+1,j,k}^+ - \phi_{i-1,j,k}^+)}{\Delta x_2}(\phi_{i,j,k}^+ - \phi_{i-1,j,k}^+),
\]

(19)

If the point \((i-1, j, k)\) is surface adjacent in the \( x \) coordinate with the surface between \((i-2, j, k)\) and \((i-1, j, k)\), the following formulas are used:

\[
\phi_{xx}^{(1)} = \frac{(\Delta x_1 + \Delta x_2)(\phi_{i,j,k}^+ - \phi_{i-1,j,k}^+)}{\Delta x_1}(\phi_{i,j,k}^+ - \phi_{i,j,k}^-) - \phi_{xx}^S
\]

(20)

\[
\phi_{xx}^{(2)} = \frac{\phi_{i,j,k}^+ - \phi_{xx}^S}{\Delta x_1 + \Delta x_2} - \phi_{xx}^S
\]

(21)

\[
\phi_{xx} = \beta \phi_{xx}^{(1)} + (1-\beta) \phi_{xx}^{(2)}
\]

(22)

where

\[
\beta = \begin{cases} 
\frac{\Delta x_2}{\Delta x_1} & \Delta x_2 < \frac{1}{2} \Delta x_1 \\
1.0 & \Delta x_2 \geq \frac{1}{2} \Delta x_1
\end{cases}
\]

and \( S \) is the surface point and \( \Delta x_2 \) is the step size between the surface point and \((i-1, j, k)\).
If the point \((i, j, k)\) is surface-adjacent in negative \(x\),

\[
\phi_{xx} = \frac{\phi_{i+1,j,k} + \phi_S^+ - \phi_S^- - \phi_x|_S}{\Delta x_1} \frac{1}{2\Delta x_1}
\]  

(23)

where \(\Delta x_1\) is the step size between point S and \((i, j, k)\).

### 2.3.3 Difference Quotients for \(\phi_{xx}, \phi_{x\theta}, \text{and } \phi_{\theta x}\)

For subsonic flow, central-difference formulas are used for accuracy. The formulas are obtained by setting

\[
\phi_{xx} = \frac{\delta(\phi_x)}{\delta x}
\]

\[
\phi_{x\theta} = \frac{\delta(\phi_x)}{\delta \theta}
\]

\[
\phi_{\theta x} = \frac{\delta(\phi_x)}{\delta \theta}
\]

and using formulas 8 or 9 with appropriate changes. As an example, for a field point that is adjacent to a surface point S and lies below S,

\[
\phi_{xx} = \frac{\Delta x_2 \phi_x|_S + (\Delta x_1^2 - \Delta x_2^2) \phi_x|_A - \Delta x_1^2 \phi_x|_B}{\Delta x_1 \Delta x_2 (\Delta x_1 + \Delta x_2)}
\]

(25)

Weighting is not necessary in these terms as the resulting difference equations are weighted (see Section 2.5.2).

The difference quotients for the cross-derivatives are also switched between central and upwind formulas depending on flow Mach number. The switching procedure is somewhat more complicated because the coefficients of these derivatives do not necessarily switch signs anywhere.
The upwind-difference quotients for cross-derivatives are relatively simple, but there are many variations depending on flow direction. Representative examples of upwind differencing for a field point that is regular with \( \phi_x > 0 \) and \( \phi_y > 0 \) (fig. 4a) are

\[
\phi_{xr} = \frac{\phi_{1-1,j,k}^+ - \phi_{1,j,k}^+ - \phi_{1,j-1,k}^+ + \phi_{1-1,j-1,k}^+}{\Delta x \Delta r}.
\] (26)

For the configuration of figure 4b,

\[
\phi_{xr} \bigg|_A = \frac{\phi_{1-1,j,k}^+ - \phi_{1,j,k}^+}{\Delta x_1} - \frac{\phi_{1,j-1,k}^+ - \phi_{1-1,j-1,k}^+}{\Delta x_2}.
\] (27)

For the configuration of figure 4c,

\[
\phi_{xr} \bigg|_A = \frac{\phi_{1-1,j,k}^+ - \phi_{1,j,k}^+}{\Delta x_1} - \frac{\phi_{1,j-1,k}^+ - \phi_{1-1,j-1,k}^+}{\Delta x_2}.
\] (28)

Although, the configuration shown in figure 4d is unlikely since supersonic flow is shown coming out of the surface, it can happen under unusual circumstances, particularly for very coarse meshes. For the configuration of figure 4d,

\[
\phi_{xr} \bigg|_A = \frac{\phi_{1-1,j,k}^+ - \phi_{1,j,k}^+}{\Delta x_1} - \phi_x \bigg|_{S_2}.
\] (29)

There is a mixture of new and old \( \phi \) values in this formula. The philosophy is to use new values for \( \phi \) where they are available, consistent with the line relaxation, except in the case of equations 27 and 29 which are of the form \( (\phi_x^+ - \phi_x)/\Delta r \).

The code uses central differencing if the local Mach number is less than one and upwind differencing if either velocity component is supersonic. The upwind- and central-difference formulas for cross-derivatives are weighted together between these two conditions. In summary, if \( M < 1 \),

\[
\phi_{xr} = \phi_{xr} \bigg|_{\text{central differenced}}
\]
Figure 4. Example Configurations for Upwind Differencing of Cross-Derivatives
and if \( |\phi_x| > a \) or \( |\phi_y| > a \)

\[ \phi_{xr} = \phi_{xr}\big|_{\text{upwind difference}}, \]

otherwise,

\[ \phi'_{xr} = \beta \phi_{xr}\big|_{\text{central diff.}} + (1 - \beta) \phi_{xr}\big|_{\text{upwind diff.}}. \]  

(30)

where

\[ \beta_{xr} = \frac{(a^2 - u^2)(a^2 - u_r^2)}{(q^2 - u^2)(q^2 - u_r^2)}. \]

(31)

It can be seen that \( \beta_{xr} = 1.0 \), when \( M = 1.0 \) and \( \beta_{xr} = 0 \) when \( |u| \) or \( |u_r| = a \).

### 2.3.4 SPECIAL \( \theta \)-DIFFERENCE QUOTIENTS

Special difference quotients for \( \phi_x \) and \( \phi_{x\theta} \) as derived in reference 3. These formulas are equivalent to the usual formulas to the order of \( \Delta \theta^2 \). They are considerably more accurate than the standard formulas if \( \Delta \theta \) is large (\( \Delta \theta = \pi/4 \) or \( \pi/2 \)) and \( \phi \) is of the form

\[ \phi(x,r,\theta) = \phi_1(x,r) + \phi_2(x,r) \sin \theta + \phi_3(x,r) \cos \theta \]

(32)

\( \phi \) has approximately the above form in the far field for inlet computations, and in the near field if the inlet is nearly axisymmetric and centered on the axis. The advantage of the special formulas is improved accuracy for very coarse meshes. Such very coarse meshes can be desirable if a sequence of computational meshes is used for convergence acceleration. The formulas are

\[ \phi_\theta = \left[ (1 - \cos \Delta \theta \ell) (\phi_{1,j,k+1} - \phi_{1,j,k}) + (1 - \cos \Delta \theta u) (\phi_{1,j,k} - \phi_{1,j,k-1}) \right] \]

\[ \times \left[ \sin \Delta \theta \ell (1 - \cos \Delta \theta u) + \sin \Delta \theta u (1 - \cos \Delta \theta \ell) \right] \]

(33)
\[
\varphi_{\theta \theta} = \left\{ \sin \alpha \varphi_{i,j,k+1} + \left( \sin \beta + \sin \gamma \right) \left[ \frac{1}{\omega^2} \varphi_{i,j,k} \right] + \left( 1 - \frac{1}{\omega^2} \right) \varphi_{i,j,k} \right\} + \frac{1}{\omega^2} \left[ \sin \alpha \left( 1 - \cos \beta \right) + \sin \gamma \left( 1 - \cos \alpha \right) \right].
\]

The use of the above difference quotients instead of formulas 8, 9, and 17 is available as a program option.

### 2.4 SURFACE DERIVATIVES

Before the field is swept, the velocity is calculated at all surface points using all old \( \phi \) values. The velocity components \( \phi_x, \phi_y, \) and \( \phi_{\theta} \) are computed first. These are the components of velocity along constant \( x, y, \) and \( \theta \) cuts of surfaces, and, except at end points of the cuts, are computed using central-difference formulas similar to equation 8. One-sided differences are used at end points. Weighting is used to eliminate possible problems due to very closely spaced points. For any given surface point, only two of these velocity components can be computed in this manner. As an example, \( \phi_y \) cannot be computed directly for an \( x \)-intersection surface point, that is, a point created by a mesh line parallel to the axis intersecting the surface.

At any point the remaining surface-derivative can be computed from

\[
n \left( \frac{2}{r^2 + n^2} \right) \varphi_x + n \left( \frac{2}{n^2 + r^2} \right) \varphi_y + n \left( \frac{2}{r^2 + n^2} \right) \varphi_{\theta} = 0 \quad (35)
\]

if the magnitude of the direction cosine \( (n_x, n_y, \text{etc.}) \) is not very small. If the magnitude of the direction cosine associated with the velocity component to be computed is very small, the above formula can be very badly behaved since it involves the reciprocal of the direction cosine. In such a situation, the missing velocity component is calculated by interpolating along one of the two cuts through the point. The cut selected is the one for which the interpolation will be the most accurate.

After all three surface velocities have been computed for each surface point, the velocity components \( \phi_x, \phi_y, \) and \( \phi_{\theta} \) for surface points are computed from
\[
\phi_x = n_x \phi_n - n_r \sqrt{n_x^2 + n_r^2} \phi_\theta + n_\theta \sqrt{n_x^2 + n_r^2} \phi_s^r \\
\phi_r = n_r \phi_n - n_\theta \sqrt{n_r^2 + n_\theta^2} \phi_\theta + n_x \sqrt{n_r^2 + n_x^2} \phi_s^r \\
\frac{\phi_\theta}{r} = n_\theta \phi_n - n_x \sqrt{n_\theta^2 + n_x^2} \phi_\theta + n_r \sqrt{n_\theta^2 + n_r^2} \phi_s^r
\]  
(36)

where \( \phi_n = 0 \).

These velocity components are used in the calculation of the cross-derivatives, \( \phi_{xr} \), \( \phi_{r\theta} \), and \( \phi_{\theta r} \), for subsonic flow, in the surface-point boundary-condition (eq. 47) under certain circumstances, and for the surface properties printed for program output.

### 2.5 FIELD-POINT DIFFERENCE EQUATIONS

A difference equation is generated for each field point. The difference equation is generated by dividing the partial differential equation (eq. 1) into terms and then splitting each term into a coefficient and an unknown. The unknown is the second derivative in the term, except for the last term of the equation where the unknown is taken as \( \phi_r \). The coefficients involve only first derivatives and are calculated using all old values of \( \phi \). The remaining part of the difference equation is linear in \( \phi \) and theoretically could be solved simultaneously for the new potential values, \( \phi^+ \), but as stated earlier, this is not practical. To make the solution process simpler for a single sweep or iteration, the unknowns are generated using difference quotients that have a mixture of old and new \( \phi \) values.

The terms in the difference equation are listed in Table 1, including the coefficient, the unknown term, and an index to the difference quotients used in differencing for the unknown terms.

The difference equations are generated using a nonconservative approach. The terms conservative or nonconservative refer to whether or not the differencing scheme explicitly conserves mass. The principal flow region where this nonconservative scheme fails to conserve mass very well is at shocks. Other than at shocks, mass conservation can be improved by using a denser mesh. The errors caused by a failure to conserve mass exactly, as determined by comparison with experimental results, appear to be relatively small.

### 2.5.1 AXIS

Points on the axis are a special case because the potential equation expressed in cylindrical coordinates is singular on the axis. This is a problem with the coordinate system and not a physical problem with the flow. This problem is resolved by requiring 0 degrees, 90 degrees, 180 degrees, and 270 degrees to be in the \( \theta \) mesh, and using the Cartesian form of the potential equation (eq. 3) at nodes on the axis.
For nodes on the axis, the terms in the potential equation are given in Table 2. Coordinates and notation are shown in Figure 5. The points A, B, C, and D can be field or surface nodes. The differencing for $\phi_x$ and $\phi_{xx}$ is the same as for the equation in cylindrical coordinates. The differencing for $\phi_y$ and $\phi_z$ uses equation 8 or 9 with appropriate $\phi$ values and step sizes. The differencing for $\phi_{yy}$ and $\phi_{zz}$ uses difference quotients (eq. 16) with appropriate $\phi$ values and step sizes.

**Table 1. Terms and Difference Quotients for Field-Point Difference Equations**

<table>
<thead>
<tr>
<th>TERM</th>
<th>COEFFICIENT</th>
<th>UNKNOWN</th>
<th>FORMULA(S) FOR SUBSONIC FLOW</th>
<th>FORMULA(S) FOR TRANSONIC FLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>REGULAR</td>
<td>SURFACE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ADJACENT</td>
<td>ADJACENT</td>
</tr>
<tr>
<td>1.</td>
<td>$a^2 - \phi_x^2$</td>
<td>$\phi_{xx}$</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>2.</td>
<td>$a^2 - \phi_r^2$</td>
<td>$\phi_{rr}$</td>
<td>16†</td>
<td>16†</td>
</tr>
<tr>
<td>3.</td>
<td>$(a^2 \frac{\phi_\theta^2}{r^2}) \frac{1}{r^2}$</td>
<td>$\phi_{\theta\theta}$</td>
<td>17,34**</td>
<td>17,34**</td>
</tr>
<tr>
<td>4.</td>
<td>$-2\phi_x \phi_r$</td>
<td>$\phi_{xx}$</td>
<td>24*</td>
<td>24*</td>
</tr>
<tr>
<td>5.</td>
<td>$-2\phi_r \phi_\theta \frac{1}{r^2}$</td>
<td>$\phi_{r\theta}$</td>
<td>24*</td>
<td>24*</td>
</tr>
<tr>
<td>6.</td>
<td>$-2\phi_\theta \phi_x \frac{1}{r^2}$</td>
<td>$\phi_{\theta x}$</td>
<td>24*</td>
<td>24*</td>
</tr>
<tr>
<td>7.</td>
<td>$(a^2 + \phi_\theta^2) \frac{1}{r}$</td>
<td>$\phi_r$</td>
<td>8†</td>
<td>9†</td>
</tr>
</tbody>
</table>

* USES ALL OLD VALUES OF $\phi$
† USES ALL NEW VALUES OF $\phi$, $(\phi^\dagger)$
** OPTIONAL FORMULA
Table 2. Terms for the Difference Equation in Cartesian Coordinates

<table>
<thead>
<tr>
<th>TERM</th>
<th>COEFFICIENT</th>
<th>UNKNOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ((a^2 - \phi_x^2)\phi_{xx})</td>
<td>(a^2 - \phi_x^2)</td>
<td>(\phi_{xx})</td>
</tr>
<tr>
<td>2. ((a^2 - \phi_y^2)\phi_{yy})</td>
<td>(a^2 - \phi_y^2)</td>
<td>(\phi_{yy})</td>
</tr>
<tr>
<td>3. ((a^2 - \phi_z^2)\phi_{zz})</td>
<td>(a^2 - \phi_z^2)</td>
<td>(\phi_{zz})</td>
</tr>
<tr>
<td>4. (-2\phi_x \phi_y \phi_{xy})</td>
<td>(-2\phi_x \phi_y)</td>
<td>(\phi_{xy})</td>
</tr>
<tr>
<td>5. (-2\phi_y \phi_z \phi_{yz})</td>
<td>(-2\phi_y \phi_z)</td>
<td>(\phi_{yz})</td>
</tr>
<tr>
<td>6. (-2\phi_z \phi_x \phi_{zx})</td>
<td>(-2\phi_z \phi_x)</td>
<td>(\phi_{zx})</td>
</tr>
</tbody>
</table>

The cross-derivatives for subsonic flow are calculated as follows:

\[
\phi_{xy} = \frac{\partial}{\partial y}(\phi_x),
\]

\[
\phi_{yz} = \frac{1}{2} \frac{\partial}{\partial y}(\phi_z) + \frac{1}{2} \frac{\partial}{\partial z}(\phi_y),
\]

\[
\phi_{zx} = \frac{\partial}{\partial z}(\phi_x).
\]

As an example,

\[
\frac{\partial}{\partial y}(\phi_z) = \frac{\Delta y_2^2 u_0|_A + (\Delta y_1^2 - \Delta y_2^2) \phi_z|_{1,1,1} + \Delta y_1^2 u_0|_B}{\Delta y_1 \Delta y_2 (\Delta y_1 + \Delta y_2)} \tag{38}
\]
where \( u_\theta |_A, u_\theta |_B \) and \( \phi_x |_{i,1,1} \) have been previously calculated.

A typical supersonic-flow difference quotient for \( \phi > a \) and \( \psi > a \) is

\[
\phi_{yz} = \frac{\phi_{i,1,1} - \phi_{B}}{\Delta y_2} + u_\theta |_B.
\]

(39)

Changes are made in the differencing at the axis if flow symmetry planes are used. If the flow is symmetrical about the plane defined by \( \theta = 0 \) degrees or 180 degrees, \( \phi_z \) equals zero on the plane of symmetry, and the potential equation for \( \theta = 0 \) degrees or 180 degrees becomes

\[
(a^2 - \phi^2_x)\phi_{xx} + (a^2 - \phi^2_y)\phi_{yy} + a^2\phi_{zz} - 2\phi_x\phi_y\phi_{xy} = 0.\]

(40)

Figure 5. Notation for the Axis
The difference quotient $\phi_{zz}$ is obtained by setting $\phi_D = \phi_C$ and $\Delta z_2 = \Delta z_1$. If the plane defined by $\theta = \theta_\xi$ is also a plane of symmetry, then $\phi_y$ and $\phi_z$ must be equal to 0.0 at the axis, and at the axis the potential equation is

$$
(a^2 - \phi_x^2)\phi_{xx} + a^2\phi_{yy} + a^2\phi_{zz} = 0,
$$

(41)

and it can be shown that

$$
\phi_{zz} = \left(\phi_{\xi \xi} - \cos^2 \theta_\xi \phi_{yy}\right) / \sin^2 \theta_\xi,
$$

(42)

where $\xi = r$ for $\theta = \theta_\xi$.

### 2.5.2 Modified Difference Equation for Field Points Adjacent to Surface

The finite-difference equation for a field point next to one or more surface points can cause numerical problems if the spacing to adjacent surface points is small relative to the local mesh spacing. The analysis uses the flow equation at the field point when the field point is not close to surface points. An interpolation equation is used to obtain the field-point $\phi$ value when the surface and the field point are very close. This is accomplished by the following formulas which weight between the flow equation and interpolation formulas depending on the step size(s) between the field point and the surface point(s).

The weights $\beta_x$, $\beta_r$, and $\beta_\theta$ for the x, r, and $\theta$ interpolation formulas, respectively, are computed by using the following relationships. The equations presented are based on the configuration of figure 3c. Cubic interpolation is used.

$$
\beta_{1x} = \begin{cases} 
1 - \frac{2\Delta x_1}{\Delta x_0} & \text{if } 2\Delta x_1 < \Delta x_0 \\
0 & \text{if } 2\Delta x_1 \geq \Delta x_0 \\
0 & \text{if the field point is not adjacent in x to a surface point}
\end{cases}
$$

(43)

The formulas for $\beta_{1r}$ and $\beta_{1\theta}$ are similar.

$$
\beta_x = \beta_{1x} - \frac{1}{2} \beta_{1x} (\beta_{1r} + \beta_{1\theta}) + \frac{1}{6} \beta_{1x} \beta_{1r} \beta_{1\theta}.
$$

(44)
The formulas for $\beta_x$ and $\beta_\theta$ are similar.

The interpolation equation for $x$ interpolation is

$$
\phi_A^+ = \eta^2(3-2\eta)\phi_S^+ + (1-\eta)^2(1+2\eta)\phi_B^*
$$

$$
- \eta(1-\eta) \left[ \eta \phi_n |_S - (1-\eta) \phi_n |_B \right]
$$

where

$$
\eta = \frac{\Delta x_2}{(\Delta x_1 + \Delta x_2)}
$$

and

$$
\phi_n = \left( \Delta x_1 + \Delta x_2 \right) \phi_x
$$

(45)

and $\phi_B^*$ is either $\phi_B$ or $\phi_B^+$ if a new value of $\phi$ is available at $B$.

The final difference equation for $\phi$ at the field point $A$ is

$$
(1 - \beta_x - \beta_r - \beta_\theta) \text{ (Difference representation of the potential equation)}
+ C\beta_x \text{ (x-interpolation equation)}
+ C\beta_r \text{ (r-interpolation equation)}
+ C\beta_\theta \text{ (\theta-interpolation equation)} = 0
$$

(46)

where $C$ is the coefficient of $\phi_A^+$ in the regular-difference flow equation for point $A$.

### 2.6 SURFACE-POINT DIFFERENCE EQUATIONS

The boundary condition at solid surfaces is that there is no flow through the surface, which requires that the velocity normal to the surface, $\phi_n$, be zero. The velocity, $\phi_n$, can be expressed in terms of its components and the direction cosines ($n_x, n_r, n_\theta$) of the unit surface normal, $n$,

$$
\phi_n = n_x \phi_x + n_r \phi_r + n_\theta \phi_\theta
$$

(47)

Referring to figure 6 for notation, the difference equation for the surface boundary condition for a surface point created by a radial mesh line intersecting the surface (r-intersect surface point) is
\[
\frac{n_x}{\Delta x} \left[ \frac{\phi_s^+ - \phi_{P''}^-}{\phi_x^-|_{P''}^-} \right] + n_r \left[ \frac{\beta}{\Delta r_1^+} \left( \frac{\phi_A^+ - \phi_s^+}{\phi_r^-|_A^-} \right) \right] \\
+ (1-\beta) \left[ \frac{\phi_B^+ - \phi_s^+}{\phi_x^-|_{B^-}^-} \right] + \frac{n_\theta}{r} \left[ \frac{\phi_s^+ - \phi_{P''}^-}{\phi_{\theta}|_{P''}^-} \right] \\
- \phi_n = 0,
\]

where

\[
\beta = \begin{cases} 
\left( \frac{2\Delta r_1}{\Delta r_0} \right)^2 & \text{if } 2\Delta r_1 < \Delta r_0 \\
1 & \text{if } 2\Delta r_1 \geq \Delta r_0
\end{cases}
\] (48)

**Figure 6. Nomenclature for Typical Surface-Point Boundary-Condition Equation**
Currently $\phi_n$ is taken as zero, however, the difference equation is expressed in terms of $\phi_n$ anticipating a later version of the code accounting for boundary-layer displacement effects by using appropriate $\phi_n$ values.

The values of $\phi$ at the points $P'$ and $P''$ are obtained by cubic interpolation using the values of $\phi$ and $\phi_r$ at points $P_0$ and $P_2$. As an example,

$$
\phi_{P'} = \eta^2(3-2\eta)\phi_{P_0} + (1-\eta)^2(1+2\eta)\phi_{P_2}, \\
- \eta(1-\eta) \left[ \eta \phi_{P_0} - (1-\eta)\phi_{P_2} \right] 
$$

(49)

where

$$
\eta = \Delta x / \Delta x_0
$$

and

$$
\phi_n = - \Delta x \phi_r
$$

The values of $\phi_\theta$ at $P'$ and $\phi_x$ at $P''$ are obtained by linear interpolation between points $P_0$ and $P_2$. As an example,

$$
\phi_\theta' = \eta \phi_\theta|_{P_0} + (1-\eta)\phi_\theta|_{P_2}, \\
(50)
$$

The point $P_0$ may be a field or surface node depending on geometry.

If the point $P'$ or $P''$ does not lie in the flowfield (i.e., is interior to the surface), equation 48 is modified by using the value for $\phi_\theta$ or $\phi_x$ (for $P'$ or $P''$ respectively) computed by differencing along the surface. (See Section 2.4.)

The formulas for $\theta$ or $x$ surface intersections are similar.

If the surface point is also a field-point mesh node, $P'$ and $P''$ are field points, point B is not used, and $\beta = 1.0$. If the surface point is on the axis or adjacent to the axis, some modifications to the formulas are made.

2.7 SOLUTION PROCEDURE

For problems such as this one which require significant quantities of computer resources (central-processor time, memory, etc.), it is essential that the code be organized for maximum efficiency. This includes such aspects as organizing the computation to minimize the amount of information transferred and the number of times information is transferred between core and disk memory. This can be done by
careful planning of the sequence of computations as well as reducing memory requirements to a minimum. Computer central-processor time requirements can be greatly affected by the sequence in which order independent quantities are calculated.

2.7.1 CODE ORGANIZATION

One of the major problems with three-dimensional codes is storage in the computer. Storing a three-dimensional field when using a typical mesh requires a large number of words. For example, an inlet mesh with \(x\), \(r\), and \(\theta\) dimensions of 81 x 41 x 16, respectively, requires 53136 locations for the potential function alone. If a copy of the potential function at a previous step is required for extrapolation or other convergence acceleration procedures, another 53136 locations are required. Any geometric information requires additional storage. Since these numbers are large compared to what is available, even for the vector computers such as the CYBER 203, it is essential that the scheme used for storing and accessing geometrical information be designed to minimize storage requirements. Disk storage may be used to provide more memory, but there are large time penalties for using disk storage; thus it is desirable to minimize its use.

Storing a flag or index for each field point so that the code can determine if the field point is next to the surface, and if so, which surface point, requires at least one storage location for each field point. This type of scheme is conceptually simple, but inefficient. To minimize storage and maximize the percentage of vector code, an inverse scheme is used in this code. Surface-affected field points are processed by sweeping through the surface points. A bit array rather than a full word (64 bits) array is used to signal whether grid points are in the flowfield or not. A bit array is used for each of the three coordinates to mark whether field points are surface-adjacent in either or both directions in that coordinate. Total storage for the four bit arrays is 4/64 of 53136 or 3321 words. The bit arrays are used to inhibit storage of contributions to coefficients from regular field-point formulas when the field point is irregular (surface-affected). The correct contributions to coefficients of the finite-difference equations are then generated and stored by sweeping through the surface points.

The geometric information stored for each surface point includes the index of the adjacent field point.

Code efficiency on a vector machine is improved by ensuring that most computation is done in a vector mode. On the CYBER 203, longer vectors (up to a maximum length of 65536) give higher computation rates. However, most vector computation involves intermediate results which are also vectors and must be stored. Hence there is a trade between the additional memory required for the intermediate vectors and the increased computation rates obtained using longer vectors. For this problem, computing coefficients for one axial plane at one time appears to be a reasonable tradeoff.

Surface points are grouped by type, adjacent in \(x\), adjacent in \(r\), adjacent in \(\theta\), or both a surface point and a field point, for each axial plane (plane perpendicular to axis). Quantities needed for coefficient calculation are gathered for all surface points of the same type, the coefficients are then calculated as a group using vector arithmetic, and the contributions to the coefficients are then scattered to the correct field-point.
equations. This maximizes the number of vector calculations and minimizes the storage of geometric information. The primary penalty is the inefficiency of the gathering and scattering operations on the CYBER 203. The gather/scatter operations are more efficient on some other computers, and on the CYBER 205 which is the successor to the CYBER 203.

2.7.2 PROCEDURE FOR SWEEPING

The sequence of computations for taking a single relaxation sweep, or iteration, for a given mesh follows:

I. The three components of velocity are calculated for all surface nodes.

II. The field is swept one axial plane (constant x) at a time in the direction of increasing x.
   A. The three components of velocity at all field nodes are calculated for the axial plane ahead of the one being processed. (The velocities are saved for the plane being swept and the planes adjacent to the plane being swept.)
   B. The coefficients for the surface-node difference equations for all surface nodes on or adjacent to the plane being processed are calculated.
   C. The coefficients for the terms in the partial differential equation are calculated for all field nodes.
   D. The coefficients of the difference equation for all field-point nodes are calculated.
   E. The surface-node equations are used to eliminate the values of $\phi$ at surface nodes from the field-point difference equations.
   F. The first radial line of difference equations is solved including the value for $\phi$ on the axis.
   G. The new value of $\phi$ on the axis is used in the solution of all remaining radial lines.
   H. The new $\phi$ values at field nodes are substituted into the surface-point difference equations and new values of $\phi$ are obtained at all surface nodes.

III. The entire field has been updated and a decision is made on whether to continue sweeping, extrapolate, change meshes, or stop.

2.7.3 CONVERGENCE ACCELERATION

The basic method described above is stable and convergent, but the rate of convergence for dense meshes can be very slow and computer times significant. There is substantial benefit from any procedure that will speed convergence and decrease the cost of a solution. Analysis and code development have been structured to facilitate the use of various techniques for convergence acceleration. One technique currently used is extrapolation of solutions as described in reference 6. The theory for
this technique can be obtained by approximating the solution process as a linear one and looking at the eigenvalues of a matrix. The number of eigenvalues is of the order of the number of field points. This approach behaves best when the eigenvalues are discrete, which is more likely when there are few of them. As a consequence, this technique works best for very coarse meshes and much less satisfactorily for fine meshes.

If \( \phi^{(n)} \) is \( \phi \) at iteration \( n \), and \( \| \cdot \| \) is a norm, then

\[
\lambda = \frac{\| \phi^{(n)} \|}{\| \phi^{(n-1)} \|}
\]

and the extrapolation formula is

\[
\phi_{\text{extrapolated}} = \frac{1}{1 - \lambda} \left( \phi^{(n)} - \phi^{(n-1)} \right) + \phi^{(n-1)}.
\]

In practice, two norms are calculated after each sweep

\[
\| \phi^{(n)} \|_1 = \sum_{\text{all field nodes}} |\phi_{i,j,k}^{(n)} - \phi_{i,j,k}^{(n-1)}|
\]

and

\[
\| \phi^{(n)} \|_2 = \max_{\text{all field nodes}} |\phi_{i,j,k}^{(n)} - \phi_{i,j,k}^{(n-1)}|
\]

where the maximum is taken over all field nodes.

Extrapolation occurs whenever the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) corresponding to \( \| \cdot \|_1 \) and \( \| \cdot \|_2 \) are steady and approximately the same for several sweeps. If they are reasonably close to being constant for several sweeps, it is assumed that the problem is behaving linearly and an extrapolation of the field will be of benefit.

Another technique for convergence acceleration is initial convergence on a coarse mesh to obtain an approximate solution, continuation of the convergence on a medium-density mesh to obtain a more accurate approximation of the solution, and then final convergence on a fine mesh. The number of mesh for each coordinate is doubled for every successive mesh. There is approximately a factor of 16 change in
convergence rate per sweep, in terms of computer central-processor time, between successive meshes. This arises due to a factor of eight change in number of mesh points, and a factor of two in the rate of $\phi$ change per sweep. Once the solution nears convergence on one of the coarse meshes, there is no further gain in continuing relaxation on the mesh because of the large truncation errors. For the meshes used in the calculations of this report, there was about one-third decrease in run time using this approach. The reason for such a small improvement is the very slow convergence on the final mesh. It takes half as many sweeps to improve the solution from coarse meshes on the final mesh as it would to do the entire calculation using the finest mesh. Very good convergence on a coarse mesh takes less than 100 sweeps. Good convergence using the fine mesh exclusively takes about 400 sweeps. Use of a still finer mesh starts to become very impractical due to the number of sweeps required and the time per sweep.

Parametric cubic interpolation as described in appendix B is used to interpolate the solution from the current mesh to the next finer mesh.

A further refinement on convergence acceleration is the multilevel technique described by Brandt(8). This procedure uses coarse meshes to resolve the errors determined on finer meshes. The number of sweeps required on the finest mesh should be small (on the order of ten to thirty) and independent of the fineness of the mesh. The majority of sweeping would be on the coarsest mesh. As can be seen from the discussion of the preceding paragraph this technique becomes better and better, relative to the existing approach, as the meshes become denser. This analysis has been established with the anticipation of implementing a multilevel procedure in the future.

2.8 GEOMETRY SPECIFICATION

The basic analysis procedure described has no inherent limitations on kinds of geometries that can be analyzed, but any implementation on the computer does have limitations. Two sources of limitations are the number of mesh that can be used and the boundary conditions that can be applied. A limitation independent of the flow analysis code is the user's ability to specify the geometry and extract the surface/mesh intersections.

The computer code for this analysis requires that all intersections of the mesh with the geometry, plus components of the surface normals, be input. The flow code then orders the points and makes some checks for completeness and consistency. The actual values could come from any source; no particular order is assumed by the flow code. The three-dimensional geometries used as samples in this report were described by a geometry system(9) which breaks the surface into subsections or patches. The coordinates of the surface on the patch are specified by parametric bicubics. This leads to an explicit specification of the surface. A separate program exists to intersect the surface so described, with a computational mesh and to generate the coordinates and surface normals at the mesh intersections.

2.9 RESULTS

Comparisons between the analysis and experiment have been made for two asymmetric geometries. In addition, a mixer-lobe geometry has been analyzed to verify the program's general three-dimensional capabilities, although no experimental results are available for comparison. The first comparison between
theory and experiment is for an asymmetric tilt-nacelle V/STOL airplane inlet, tested in the NASA-Ames Research Center's 40- by 80-ft wind tunnel. Figure 7 presents a graphic display of the inlet and spinner geometry. The geometry description used in the analysis is smooth and the entire inlet is specified. Only half the inlet is shown in figure 7, and curves are represented using linked straight-line segments due to limitations of the plotting equipment. The test and results of the test are described in reference 10. Tabular data from the test is found in reference 11.

Predictions of the analysis have been compared with experimental measurements for three test cases (figs. 8, 9, and 10). There are two comparisons for an angle of attack of 60 degrees. The second comparison is for a greater airflow and freestream Mach number than the first. The last comparison is for an angle of attack of 90 degrees. The analysis predicts the experimentally measured Mach number distribution very closely. There is a slight, but consistent, under-prediction of the peak Mach number. This is possibly due to grid density and is discussed later. The only poor prediction is for the external lee side for 90 degree angle of attack (fig. 10). This probably is an interference or boundary-layer effect in the experiment.

Figure 7. Graphical Display of V/STOL Airplane Inlet Geometry
Figure 8. Cowl Surface Mach Number Distribution for an Asymmetric VISTOL Airplane Inlet
Figure 9. Cowl Surface Mach Number Distribution for an Asymmetric VISTOL Airplane Inlet
Figure 10. Cowl Surface Mach Number Distribution for an Asymmetric VI/STOL Airplane Inlet
A series of comparisons have been made for a typical commercial-transport turbofan-engine inlet design. It has an asymmetric lip section varying both in section shape and contraction ratio. The contraction ratio on the lee side is 1.246 and on the windward side 1.28. The inlet was designed with an axisymmetric diffuser. The inlet centerline is tilted down five degrees with respect to the engine centerline and a short transition region exists between the inlet and the compressor (fan) face. This tilt of the inlet centerline is to reduce cruise drag by aligning the inlet with the local flowfield while still placing the engine and hence its thrust vector in the desired orientation. The inlet flowfield was computed using engine-centerline oriented coordinates, and as a consequence, the inlet appears very asymmetric. A cross-section of the inlet along with a computational mesh is shown in figure 11. Dimensions are in meters for the full-scale inlet.

A 0.16 scale model of the inlet has been tested* in the Boeing 9- by 9-ft low-speed propulsion wind tunnel. The same inlet configuration in a 0.47 scale was tested(12) in the NASA-Ames Research Center’s 40- by 80-ft wind tunnel. The NASA-Ames test model was modified slightly in the vicinity of the fan face to attach to a different engine than that for which the inlet was originally designed. The Boeing 9- by 9-ft tunnel test used a suction source instead of an engine at the fan face. The 9- by 9-ft tunnel test gave results at greater airflows than could be obtained in the Ames test. The comparisons between analysis and experiment for the 0.16 scale model are shown in figures 12 and 13. Figures 14 and 15 show results for the 0.47 scale model. The most significant discrepancy between the results of the analysis and the experimental measurements is for the greatest peak Mach number (fig. 13). Some of the discrepancy is due to blockage effects of a thick boundary layer in the diffuser on the windward side. This condition is near where the inlet boundary layer separates. The analysis curves are drawn to the compressor station. As the inlet is tilted relative to the engine, each curve ends at a different inlet centerline station.

A single lobe of a mixer has been analyzed with this code. The lobe is shown in figure 16. This is a shaded-graphic representation(13) of the upper surface of the mixer. The inner surface was a constant-diameter cylinder. The lobe extended from 0 degrees to 45 degrees and represents one lobe of an eight-lobe jet-engine mixer nozzle. Half a lobe could have been analyzed by taking advantage of the center plane of symmetry. The mixer shape was developed as a computer-program test case and not an actual mixer. It is shown to illustrate possible applications of the code. As there is no experiment with which to compare, the computed results are not shown. Use of the code for configurations other than inlets should include code validation by comparison of predictions with experiments for such configurations.

2.10 ACCURACY

The comparison between analysis and experiment for the V/STOL airplane inlet (figs. 8, 9, and 10) show almost perfect agreement except at the point of peak Mach number on the windward side of the inlet. The progression of peak Mach number values for the case of figure 9 is 0.743, 0.893, and 1.007 for the three mesh levels used, clearly indicating a trend of an increase in peak Mach number for finer meshes. It is believed that the fine mesh used is the coarsest that will yield adequate accuracy for inlet

*Boeing data, unpublished.
Figure 11. Computational Flowfield and Mesh in the Vicinity of a Commercial-Transport Turbofan-Engine Type Inlet
Figure 12. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet
### Figure 13. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet

**ANALYSIS**

<table>
<thead>
<tr>
<th>CIRCUMFERENTIAL ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0° (LEE SIDE)</td>
</tr>
<tr>
<td>90°</td>
</tr>
<tr>
<td>180° (WINDWARD SIDE)</td>
</tr>
</tbody>
</table>

**ANGLE OF ATTACK = 20° WITH RESPECT TO INLET CENTERLINE**

**FREESTREAM VELOCITY = 90 m/s**

**M_C.F. = 0.64**

**M, SURFACE MACH NUMBER**

**x, DISTANCE FROM HIGHLIGHT ALONG INLET CENTERLINE, FULL SCALE, METERS**

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>EXP.</th>
<th>CIRCUMFERENTIAL ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0° (LEE SIDE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90°</td>
</tr>
<tr>
<td></td>
<td></td>
<td>180° (WINDWARD SIDE)</td>
</tr>
</tbody>
</table>
Figure 14. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet
Figure 15. Cowl Surface Mach Number Distribution for an Asymmetric Commercial-Transport Turbofan-Engine Type Inlet

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>EXP.</th>
<th>CIRCUMFERENTIAL ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0° (LEE SIDE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90°, 270°</td>
</tr>
<tr>
<td></td>
<td></td>
<td>180° (WINDWARD SIDE)</td>
</tr>
</tbody>
</table>

ANGLE OF ATTACK = 30° WITH RESPECT TO INLET CENTERLINE
FREESTREAM VELOCITY = 75 m/s
M_C.F. = 0.40

x, DISTANCE FROM HIGHLIGHT ALONG INLET CENTERLINE, FULL SCALE, METERS
Figure 16. View of Upper Surface of a Mixer Lobe
design and analysis. The mesh used is an excellent tradeoff between cost and accuracy, since the solution for any significantly denser mesh would be extremely expensive to compute. It does appear that use of a finer mesh would result in higher predicted peak Mach numbers and slightly better agreement with experiment. Such calculations will be made when the multilevel technique of Brandt is implemented.

There is an additional check on solution accuracy built into the program. The computed velocity and density profiles in the inlet are integrated at each axial station to compute mass flow, and the error relative to the mass flow enforced as a boundary condition is determined. Typical maximum errors are one-half to one-and-one-half percent. The magnitude of the error is a function of the complexity of the geometry, the mesh used, the degree of convergence, and the strength of any internal shocks.

Overall, the program appears to predict inviscid flowfields to the accuracy that they can be measured experimentally. At flow conditions where Mach number gradients are large and boundary layers are thick, or shock-wave/boundary-layer interactions are present, agreement with experiment is not as good. It is expected that use of denser meshes and more stringent convergence tolerances would slightly improve agreement with experiment at the cost of significantly increased computer-time requirements. Improved accuracy for flows with significant viscous effects would require that viscous (boundary-layer) corrections be made to the inviscid flowfield.

2.11 CONCLUSIONS

The analysis technique described is capable of quite acceptable prediction for aircraft design and analysis purposes of flowfields for single-object geometries with essentially inviscid, irrotational flow.

Reasonable predictions can be made for flows which are primarily irrotational except for thick, but attached, boundary layers. Geometry limitations are primarily due to the number of mesh lines the computer code can use and still have reasonable computation times. The restriction is that any major geometric variations have to have a scale of several mesh spacings. The restriction to inviscid irrotational flow is simply because viscous or boundary-layer effects have not been incorporated into the flow analysis. If these effects are significant, such as occurs with very thick or separated boundary layers, predictions will be poor.
3.0 USE OF THE PROGRAM

This section discusses the use of the program exclusive of the generation of the geometry file. A detailed description of the input formats along with sample input and output files, and a troubleshooting guide, are presented. Also included is a discussion of selection of the computation mesh.

The problem of specifying the geometry is not discussed. The specification of geometry and generation of the geometry part of the input file can be a major task. It has been treated as a separate topic with separate documentation. The code will accept geometry files from any source. The program does not require any special order to the points in the geometry file.

The code, in its present form, can handle inlets with or without centerbodies, three-dimensional bodies, ducts, and bodies or inlets in a duct (e.g., wind tunnel). Specific limitations on geometries relative to mesh placement are discussed.

3.1 COMPUTATIONAL PARAMETERS

The analysis is programmed in CDC CYBER 200 FORTRAN language 1.4(14) for the Control Data Corporation CYBER 203 computer. The code makes extensive use of the vector extensions of this language and is explicitly vectorized. Use is made of most of the million-word memory of the CYBER 203 at the NASA-Langley Research Center.

A typical fine mesh for an inlet calculation is 69 axial-mesh by 37 radial-mesh by 16 circumferential-mesh or 40848 points. The calculation of a solution, using just the fine mesh, to reasonable convergence takes approximately 400 sweeps and seven to eight minutes of central-processor time. Essentially the same solution can be obtained using a sequence of three grids in approximately five- and one-half minutes. A reasonable convergence pattern is 100 sweeps on the coarsest mesh (18 by 10 by 16), 150 sweeps on the intermediate mesh (35 by 19 by 16), and 200 sweeps on the fine mesh. Typical times per sweep are 0.18 seconds per sweep on the coarse mesh, 0.42 seconds per sweep on the intermediate mesh, and 1.1 seconds per sweep on the fine mesh. Some additional time is required for setting up geometry parameters, mesh changing, and printing the solution. Solutions have been calculated using a sequence of 4, 8, and 16 circumferential meshes, but the coarsest mesh had a Δθ of 90 degrees and it is possible for this coarse spacing to cause some problems. Consequently, better convergence is presently obtained by keeping the number of θ mesh constant at 16. Run time is slightly dependent on the number of supersonic field points that have to be calculated.

3.2 INPUT FORMAT

The first two cards of the input deck are title cards and are printed at the start of the output for identification purposes. All input except the title cards is by means of order independent groups headed by keywords. The purposes of this particular input format are to allow certain groups to be optional, make the input file more readable, and to facilitate checking of input data by the program. Certain input groups are mandatory, and others are optional and may be omitted. All input, except for title cards, consists of numbers or words (depending on group) in fields of 10 columns wide, maximum of six fields per input line. All numbers are floating point and require a decimal point. Only the first four characters of keywords are checked.
There are certain interrelations among various input groups that have to be taken into account. If convergence is to be obtained on a sequence of meshes, the number of \( x, r, \) and \( \theta \) mesh can have only certain values. This is because coarser meshes are formed by deleting every other mesh line. Also, a compressor face, if there is one, must lie on an \( x \) mesh belonging to the coarsest mesh. There are restrictions on number of mesh and number of surface points relating to declared array lengths in the computer code.

The program has the capability to use up to three mesh-density levels to provide more efficient convergence. The number of levels is controlled by the SWEEPS option. The mesh and geometry for the finest mesh level must be input. Coarser meshes for \( x \) and \( r \) are formed by deleting exactly every other mesh from the previous mesh. This places restrictions on the number of mesh allowed in the finest mesh, as the first and last mesh line have to remain when every other mesh is deleted. The \( \theta \) mesh is a special case. There is an option to control the manner in which the \( \theta \) mesh is varied between levels. The number of \( \theta \) mesh can be held constant for two successive levels, or every other \( \theta \) mesh value can be deleted for the coarser mesh.

The program allows the use of planes of symmetry to cut the number of mesh needed to make a calculation. If the largest \( \theta \) mesh value input is 180.00 degrees, the plane 0 degrees to 180 degrees is taken to be a plane of symmetry. If the largest \( \theta \) mesh input is less than 180 degrees the flow is assumed symmetrical about 0 degrees and the largest \( \theta \) value input. Zero degrees must always be input as a \( \theta \) mesh.

### 3.2.1 INPUT-GROUP SUMMARY

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREEstream</td>
<td>Speed of sound, freestream velocity, angle of attack, and angle of yaw</td>
</tr>
<tr>
<td>XMESh</td>
<td>Axial mesh values</td>
</tr>
<tr>
<td>RMESh</td>
<td>Radial mesh values</td>
</tr>
<tr>
<td>TMESh</td>
<td>Circumferential mesh values</td>
</tr>
<tr>
<td>GEOMetry</td>
<td>Surface/mesh intersections: coordinates and surface normals values</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPressor</td>
<td>Indicates an inlet geometry and specifies inlet mass flow</td>
</tr>
<tr>
<td>SWEEps</td>
<td>Convergence control parameters</td>
</tr>
<tr>
<td>THETa</td>
<td>Control of number of ( \theta ) planes for each mesh-density level</td>
</tr>
</tbody>
</table>
SCDIf
Indicate special $\theta$ differencing to be used

PRINt op
Requests printout of various categories of geometrical information

SFLOw
Requests surface flow variable printout at end of run

SURF pr
Requests printout of flow variables along $\theta$ constant cuts of surface

FIELd pr
Requests printout of flow variables at constant $\theta$ cuts of flowfield

IPRI
Request for printout at other than level 3 for multilevel calculation

Keyword Description

OPTIONAL (Diagnostic)

DEBUg
Requests print of coefficients, velocities and potential function for a specified axial cut and sweep number

3.2.2 INPUT-GROUP DESCRIPTIONS

FREESTREAM

This group specifies the velocity and orientation for the freestream relative to the geometry.

The scaling of the velocities is essentially arbitrary except that they should be of order one to avoid difficulties with print formats. Note that $q_{\infty}/a_{\infty} = M_{\infty}$.

Required input group, no default values.

Card 1 Cols. 1-4 'FREE' Keyword
Card 2 Cols. 1-10 AINF $a_{\infty}$, freestream speed of sound
Card 2 Cols. 11-20 QINF $q_{\infty}$, freestream velocity
Card 2 Cols. 21-30 ALPHA $\alpha$, angle of attack, degrees, arctan ($v_{\infty}/u_{\infty}$)
Card 2 Cols. 31-40 BETA $\beta$, angle of yaw, degrees, arctan ($w_{\infty}/u_{\infty}$)

Note: Input of "FREE STREAM" which is 11 characters instead of "FREE" or "FREESTREAM" will draw an error message.
**XMESH**
**RMESH**
**TMESH**

These groups handle the input of the computational mesh, x, r, and theta, theta in degrees. The values do not have to be in any order. Theta mesh must include 0, 90, 180, and 270 degrees unless there is a plane of symmetry. Zero degrees must always be included.

Required input groups, no default values.

Card 1   Cols.  1-4   'XMES' or 'RMES' or 'TMES'
          or 11-20   NX
          or 11-20   NR
          or 11-20   NT  Number of mesh values to be read, six per card.

Card 2   Cols.  1-10  MESH(1)   Axial, radial or circumferential location of mesh,
          11-20   MESH(2)    six values per card, as many cards
          21-30   . . .    as required. Theta must
          . . .    be in degrees.

Note:

1)  \(NX*NR*NT\leq56000\)

2)  \(NR*NT\leq800\)

3)  \(NX*NR\leq2800\)

4)  If three mesh levels are to be used in the analysis:
    \(NX-1=4,8,12,16,\ldots,\text{ or } 100.\)
    \(NR-1=4,8,12,16,\ldots,\text{ or } 80.\)

5)  The compressor face, if there is one, must lie on a x mesh in the coarsest mesh (i.e., \(x_{CF}=x_{i}, i=1, \text{ or } 5, \text{ or } 9, \text{ or } 13 \text{ etc. for a } 3\text{-level mesh}).

**GEOMETRY**

This group consists of the coordinates of the intersections of the mesh with the geometry and the direction cosines of the surface normal at each intersect.
Required input group, no default values.

Card 1  Cols.  1-4    'GEOM'    Keyword
           11-20 NSURTOT  number of intersects

Card 2  Cols.  1-10  SURFX(1)  x
           11-20  SURFR(1)  r
           21-30  SURFT(1)  θ (degrees)
           31-40  COSX(1)  n_x
           41-50  COSR(1)  n_r
           51-60  COST(1)  n_θ

Card 3  Cols.  1-10  SURFX(2)  
           .      
           .      

Card
NSURTOT+1

COMPRESSOR (Optional)

Signals that there is an inlet geometry and specifies the Mach number at the compressor face.

Card 1  Cols.  1-4    'COMP'    Keyword
           11-20 AMACHCF  Mach number to be enforced at
                            compressor face.

SWEEPS (Optional)

This group controls the sweeping process by allowing control of the number of mesh-density levels, the maximum number of sweeps on each mesh-density level and a convergence criteria for each mesh-density level.

Default is a three-level mesh with default values listed.

Card 1  Cols.  1-4    'SWEE'    Keyword

Card 2  Cols.  1-10  NSWPM(1)  Maximum number of sweeps
           11-20  NSWPM(2)  on coarsest mesh (or total
                            if one-level calculation)
           21-30  NSWPM(3)  Maximum number of sweeps
                            on level 2, level 3 for a
                            two-level calculation (zero
                            implies a single-level calculation)
Card 3 Cols. 1-10 CONVT(1) Convergence parameter, change in $\phi$ times $10^6$. Sweeping on a level will stop when $|\Delta \phi|/(\phi_{\text{max}}-\phi_{\text{min}}) \times 10^6 \leq \text{CONVT}(i)$

CONVT(i) corresponds to the same level as NSWPM(i)

Default values: 3 levels
NSWPM(1) = 200.
NSWPM(2) = 200.
NSWPM(3) = 200.

CONVT(1) = 1.0
CONVT(2) = 1.0
CONVT(3) = 1.0

THETA (Optional)

This group controls the number of $\theta$ grid used for each mesh-density level. The number of levels is controlled by either the SWEEPS option or the default value (three levels). The number of $\theta$ intervals can be held constant or doubled for any mesh change. The values for the number of $\theta$ grid must be consistent with the symmetry flag.

Default for this option is no change in $\theta$ grid for different levels.

Card 1 Cols. 1-4 'THET' Keyword
Card 2 Cols. 1-10 NT, Number of $\theta$ grid for coarsest mesh-density level
11-20 NT, Number of $\theta$ grid for level 2
21-30 NT, Number of $\theta$ grid for level 3

Note that if there is a plane of symmetry (less than 0 degrees to 360 degrees geometry input)

$$ NT_i = NT_{i+1} \text{ or } NT_i = (NT_{i+1}+1)/2.0, $$

otherwise,

$$ NT_i = NT_{i+1} \text{ or } NT_i = NT_{i+1}/2.0 $$

SCDIff (Optional)

This group allows use of special $\theta$ differencing for improved accuracy with very coarse $\theta$ meshes.

Default is regular differencing.
Card 1 Cols. 1-4 'SCDI' Keyword
11-20 ANUM 0.0 Regular differencing.
1.0 Special \( \theta \) differencing
(any other value than 0.0 or 1.0 will be treated as 0.0).

PRINT OP (Optional)

Inputs any or all of a group of keywords to obtain printed output for certain geometrical quantities.

Card 1 Cols. 1-4 'PRIN' Keyword
Card 2 Cols. 1-10
11-20

SPIN_INPUT
List of the surface points in the order read.

SPORDER
List of surface points in the internal ordering used in the analysis.

SPEC_PNTS
List of special points.

TYPE2
List of Type 2 points.

MAP
Lists of \( x, r, \) and \( \theta \) constant cuts of the surface. Lists include surface-point indexes, surface-point coordinates, arc length along the cuts, and components of the surface normals.

CUTS
Lists of \( x, r, \) and \( \theta \) constant cuts of the surface. Lists include surface-point indexes, surface-point coordinates and surface velocities.

SFLOW (Optional)

This group controls printing of flow properties along the surface. Default is printing of every fourth cut for all surfaces. This default corresponds to coarse-mesh cuts for a three-level calculation.

Card 1 Cols. 1-4 'SFLO' Keyword
Card 2 Cols. 1-10 SKIPX See below.
11-20 SKIPR
21-30 SKIP T

For the fine mesh, every constant \( x \) cut will be printed if \( \text{SKIPX} = 1 \). If \( \text{SKIPX} = 0 \) no cuts will be printed. Otherwise, cuts will be printed for \( x = X(I), I = 1, 1 + \text{SKIPX}, 1 + 2*\text{SKIPX}, \) etc. \( \text{SKIPR} \) and \( \text{SKIP T} \) work the same for \( r \) and \( \theta \) constant cuts of the surfaces respectively.
SURFACE PR (Optional)

Surface properties along $\theta$ constant cuts are printed for $\theta$ equal to 0 degrees, 90 degrees, 180 degrees, and 270 degrees for all runs except those with planes of symmetry. This option allows the printing of surface properties at other $\theta$ mesh values.

Card 1  Cols.  1-4 'SURF'  Keyword
          11-20 NSURPR  Number of extra $\theta$ printout planes.

Card 2  Cols.  1-10 THET(1) $\theta$ values for extra surface-property output. Six values per card (in degrees).
           11-20 THET(2)

FIELD PR (Optional)

This option determines for which $\theta$ mesh values the field properties are to be printed.

Card 1  Cols.  1-4 'FIEL'  Keyword
          11-20 NFFPR  Number of $\theta$ values for which field properties are to be printed.

Card 2  Cols.  1-10 THETW Values of $\theta$ mesh for printing field properties. Six values per card (in degrees).
           11-20

IPRI (Optional)

This option allows printing of solution properties for the coarse meshes including mass-flow conservation computation. Default is no printout for coarse meshes.

Card 1  Cols.  1-4 'IPRI'  Keyword

DEBUG (Optional)

Diagnostic print option. Prints internal parameters for a given x mesh index and sweep number.

Card 1  Cols.  1-4 'DEBU'  Keyword

Card 2  Cols.  1-10 NPROPPR  Sweep number
           11-20 IPROPPR  x-plane index
3.2.3 GEOMETRY SPECIFICATION

Given a mesh, the user is required to provide all intersections between the mesh and the geometry to be analyzed. The information to be provided is the x, r, and \( \theta \) coordinates of the intersection, and the components of the unit normal to the surface at that intersection point, \( n_x \), \( n_r \), and \( n_\theta \). The surface normals must be oriented such that the normals point from the surface into the flow. No particular order has to be followed in providing the intersections, but the set has to be complete. The program checks that the input points are complete and self-consistent, and generates internal maps connecting the points in x, r, and \( \theta \) constant cuts.

The geometry points can come from any source. A package of procedures using parameteric bicubic patches has been developed to specify the geometry, and generate the geometry input file. It is documented separately (ref. 9).

The program has been used to calculate the flow about three-dimensional ducts, inlets and bodies. Most of the program is extremely general and the flow about other geometries could possibly be calculated with minor program modifications.

3.2.4 MESH SPECIFICATION

The selection of the computational mesh can have a very significant effect on the cost and accuracy of a calculation. This is because the mesh affects the accuracy of the final converged solution through the truncation errors, and the rate of convergence of the solution. The influence of these effects is understood only in qualitative terms. In general, squarer meshes \((\Delta x = \Delta r = r \Delta \theta)\) converge better. The critical mesh aspect ratio is \(\Delta x / r \Delta \theta\) since the use of line relaxation in the radial direction eliminates some of the effects of \(\Delta r\). As the ratio of \(\Delta x / r \Delta \theta\) departs significantly from unity, problems with convergence can occur. It is not possible, for inlet computations, to keep the far-field boundaries where they belong, keep a reasonable mesh near the inlet, and keep the mesh aspect ratio near unity everywhere. The best solution to this dilemma is to insure that any extreme aspect ratios occur in the far field where the solution changes little from the initial field.

For accuracy, very fine meshes must be used in regions of the flow where large velocity gradients are expected. Inside ducts (including inlet ducts) it is desirable to have a reasonably dense mesh with mesh aspect ratios near unity in order to have good conservation of mass by the analysis.

Table 3 gives a sample coarse mesh for a typical inlet configuration shown in figure 17. The fine meshes for x and r are generated by inserting three equally-spaced mesh lines between each of the coarse mesh lines. This will give a reasonable mesh for the inlet configuration shown. For vastly different geometries, the user may have to experiment with several possible meshes. It is very strongly suggested that the user make a sketch to scale, such as figure 17, showing the geometry and coarse mesh for any critical cuts, and take a good look at the sketch.

Mesh Selection General Rules

- Finer mesh in critical flow areas, that is areas where flow gradients are expected to be large.
Table 3. Standard Coarse Mesh for an Inlet

<table>
<thead>
<tr>
<th>x MESH</th>
<th>r MESH</th>
<th>Θ MESH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. -4.0 L</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2. -2.0 L</td>
<td>0.3 R</td>
<td>22.5</td>
</tr>
<tr>
<td>3. -1.0 L</td>
<td>0.6 R</td>
<td>45.0</td>
</tr>
<tr>
<td>4. -0.5 L</td>
<td>0.8 R</td>
<td>67.0</td>
</tr>
<tr>
<td>5. -0.2 L</td>
<td>0.9 R</td>
<td>90.0</td>
</tr>
<tr>
<td>6. -0.1 L</td>
<td>1.0 R</td>
<td>112.5</td>
</tr>
<tr>
<td>7. 0.0</td>
<td>1.2 R</td>
<td>135.0</td>
</tr>
<tr>
<td>8. 0.1 L</td>
<td>1.6 R</td>
<td>157.5</td>
</tr>
<tr>
<td>9. 0.2 L</td>
<td>2.4 R</td>
<td>180.0</td>
</tr>
<tr>
<td>10. 0.4 L</td>
<td>4.0 R</td>
<td>202.5</td>
</tr>
<tr>
<td>11. 0.6 L</td>
<td>7.0 R</td>
<td>225.0</td>
</tr>
<tr>
<td>12. 0.8 L</td>
<td>10.0 R</td>
<td>247.5</td>
</tr>
<tr>
<td>13. 1.0 L</td>
<td></td>
<td>270.0</td>
</tr>
<tr>
<td>14. 1.2 L</td>
<td></td>
<td>292.5</td>
</tr>
<tr>
<td>15. 1.5 L</td>
<td></td>
<td>315.0</td>
</tr>
<tr>
<td>16. 2.0 L</td>
<td></td>
<td>337.5</td>
</tr>
</tbody>
</table>
Figure 17. Standard Coarse Mesh for an Inlet Flowfield Calculation
Uniform or near-uniform meshes in ducts (including inlet ducts).

- Mesh spacing should never be more than half or double between adjacent mesh (in the same coordinate).

### 3.2.5 GEOMETRY LIMITATIONS

This section is an attempt to give the user some understanding of what flowfields the code can successfully analyze, and those which it cannot. Also, it is intended to give the user some basic understanding of what the problems are when the program cannot successfully calculate the flowfield about a geometry as it may be possible to make a simple modification to the code to allow it to calculate a flowfield for which it previously was unsuccessful.

The analysis is very general with respect to bodies lying internal to the computational volume. The primary limitations internal to the computational volume are mesh density and placement. Two difficulties may occur: the code may fail completely or it may give the wrong answer. Figure 18 illustrates limitations on mesh density in a passage between surfaces. If any one of a sequence of meshes is as shown in figure 18a the code will fail. A mesh such as that shown in figure 18b will not cause the code to fail, but accuracy probably would be poor. This may not be a problem if, for example, the mesh shown is the coarsest of a sequence of three meshes as the accuracy of the final solution is primarily dependent on the density of the finest mesh. Figure 19 illustrates that the mesh must be dense enough to resolve important features of the geometry. A solution calculated using the mesh shown in figure 19a would not predict any effects due to the wavy wall. A solution calculated using the mesh of figure 19b will predict at least some of the effects due to the wavy wall.

The user has some control over placement of mesh, but certain mesh placement patterns could cause severe difficulties with convergence. Uniformly spaced mesh lines result in the best convergence behavior. A very irregular spacing of mesh lines, in an attempt to place the mesh where they are required for accuracy, may be self-defeating in that it might take a totally unreasonable number of sweeps to obtain convergence. This is a dilemma that would require numerical experiments to resolve for any given geometry.

There are limitations on the total number of mesh that can be used which are built into the computer code. These could be changed by changing the code, but if the total number of mesh is increased, the run-time of the computer code also increases, and by a significantly greater factor. This run-time problem will be substantially alleviated when a version of the code with the multilevel procedure incorporated is available.

Another limitation on the current version of the code is that there is no provision for application of a Kutta condition. While there is not believed to be any fundamental reason why a Kutta boundary-condition option could not be added to the code, it is thought to require a significant engineering effort to incorporate.

The external (edge of computational volume) boundary conditions allowed are those for an inlet, duct, body, or a body or inlet in a duct. Other geometries are possible with program modifications. The nature and degree of difficulty of such modifications would vary from problem to problem.
Figure 18. Mesh-Density Limitations Between Surfaces
Figure 19. Mesh-Density Limitations Relative to Surface Geometry
The program uses cylindrical coordinates which are reasonably efficient for shapes that are nearly bodies of revolution. It is planned to incorporate a Cartesian-coordinate option in a later version of the code. Cartesian coordinates are much more efficient than cylindrical coordinates for analyzing some flowfields.

3.2.6 SAMPLE INPUT FILES

Several examples of input files (figs. 20a and 20b) are shown in an abbreviated form in order to help clarify the input file descriptions of Section 3.2.2. The files have been shortened by deleting most of the list of intersections and normals from the input files. The input groups on the files have been arranged for convenience of display by placing the "GEOMETRY" group as the last group on each file.

The input file of figure 20a is the file used to generate the sample output of Section 3.3.

3.3 PROGRAM OUTPUT

The program output is arranged so that there is an introductory section, convergence history, and the flowfield solution. The introductory section (fig. 21a) presents program title, abstract, informative messages, and a list of consultants followed by specific information about the current run. The specific run information consists of run title, a repeat of program inputs such as free-stream conditions, convergence parameters, program options selected, etc. These are followed by the mesh. Input surface points and other geometry information can be printed at the user’s option. The headings used on surface point and geometry printouts are explained in Table 4.

The convergence history (fig. 21b) includes sweep number, maximum and average residues for field points and surface points, location of maximum residue, number of supersonic points, the extrapolation parameters 1/(1-α₁) and 1/(1-α₂), and flags to indicate convergence/divergence and extrapolation. Headings for the convergence history are listed in Table 5.

Following the convergence history for each level, timing information for that level is printed (fig. 21c).

After the timing information for the last level (and optionally for other levels), surface properties for θ = 0 degrees, 90 degrees, 180 degrees and 270 degrees cuts (optionally for other θ cuts), are printed (fig. 21d). The headings for surface-property printouts are listed in Table 6. Field properties may be printed at the user’s option (fig. 21e). Headings for field-property printouts are listed in Table 7.

A mass-flow computation (fig. 21f) is printed as a check on program convergence and/or accuracy. The mass flow for a duct is computed at each x station by integrating the computed solution profiles. The calculated mass flow is compared to the mass flow specified at the exit station.

The last printout (fig. 21g) is optional (controlled by the SFLO input group) and allows printing of surface properties for x, r or θ constant cuts. Headings are explained in Table 6.
## Figure 20a. Truncated Input File For An Inlet

### Table 20: Truncated Input File For An Inlet

<table>
<thead>
<tr>
<th>Angle</th>
<th>Cosine</th>
<th>Sine</th>
<th>Phi</th>
<th>Omega</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.9848</td>
<td>0.1736</td>
<td>0.1736</td>
<td>0.9848</td>
</tr>
<tr>
<td>20</td>
<td>0.9661</td>
<td>0.2598</td>
<td>0.2598</td>
<td>0.9661</td>
</tr>
<tr>
<td>30</td>
<td>0.9462</td>
<td>0.3420</td>
<td>0.3420</td>
<td>0.9462</td>
</tr>
<tr>
<td>40</td>
<td>0.9251</td>
<td>0.4226</td>
<td>0.4226</td>
<td>0.9251</td>
</tr>
<tr>
<td>50</td>
<td>0.9032</td>
<td>0.5016</td>
<td>0.5016</td>
<td>0.9032</td>
</tr>
<tr>
<td>60</td>
<td>0.8807</td>
<td>0.5788</td>
<td>0.5788</td>
<td>0.8807</td>
</tr>
<tr>
<td>70</td>
<td>0.8571</td>
<td>0.6545</td>
<td>0.6545</td>
<td>0.8571</td>
</tr>
<tr>
<td>80</td>
<td>0.8321</td>
<td>0.7288</td>
<td>0.7288</td>
<td>0.8321</td>
</tr>
</tbody>
</table>

### Description

The table above shows the truncated input file for an inlet with various angles, cosines, sines, and other parameters. The data is useful for understanding the geometric and flow characteristics of the inlet.
Figure 20b. Truncated Input File For A Mixer Lobe
ABSTRACT -

P465 computes solutions for the complete equation for compressible potential flow. It can predict subsonic or transonic flow fields about fully three-dimensional inlets, ducts and bodies. The analysis uses cylindrical coordinates. Finite differences and successive line over relaxation (SOLOR), extrapolation and a sequence of increasingly denser mesh sizes can be used to accelerate convergence. The analysis is programmed in extended FORTRAN IV for the Control Data Corporation CYBER 203 computer.


NX = 18 35 69
NR = 10 19 37
NT = 16 16 16

MAXIMUM NUMBER OF SWEEPS = 100 150 200

CONVERGENCE TEST VALUES *(10**6)*
1.0000 1.0000 1.0000

SPECIAL SIN/COS DIFFERENCE QUOTIENTS USED FOR THETA COORDINATE

SURFACE FLOW PROPERTIES PRINT REQUESTED -

X CONSTANT CLTS X=X(I) (I-1) DIVISIBLE BY 4
R CONSTANT CUTS R=R(J) (J-1) DIVISIBLE BY 4
THETA CONSTANT CUTS THETA=THETA(K) (K-1) DIVISIBLE BY 4

IN ADDITION TO THE 0, 90, 180, AND 270 DEGREE PLANES
SURFACE POINT PRINTOUT HAS BEEN REQUESTED AT -
135.0 0 DEG.
225.0 0 DEG.

PRINTOUT OF THE FLOWFIELD HAS BEEN REQUESTED AT -
180.0 0 DEG.

NAXIRUN NLCEEC OF SYEEPS

Figure 21a. Continued
Figure 21a. Continued
Figure 21a. Concluded
<table>
<thead>
<tr>
<th>Sweep</th>
<th>Field Points</th>
<th>Surface Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.7038E-03</td>
<td>2.9142E-03</td>
</tr>
<tr>
<td>2</td>
<td>6.4244E-02</td>
<td>3.9265E-03</td>
</tr>
<tr>
<td>3</td>
<td>4.7503E-03</td>
<td>3.6476E-03</td>
</tr>
<tr>
<td>4</td>
<td>4.3254E-03</td>
<td>4.3497E-03</td>
</tr>
<tr>
<td>5</td>
<td>3.1995E-02</td>
<td>2.9861E-02</td>
</tr>
<tr>
<td>6</td>
<td>2.9661E-02</td>
<td>2.8327E-03</td>
</tr>
<tr>
<td>7</td>
<td>2.9110E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>8</td>
<td>2.9026E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>9</td>
<td>2.8436E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>10</td>
<td>2.8119E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>11</td>
<td>2.8086E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>12</td>
<td>2.7962E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>13</td>
<td>2.7832E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>14</td>
<td>2.7711E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>15</td>
<td>2.7632E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>16</td>
<td>2.7514E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>17</td>
<td>2.7405E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>18</td>
<td>2.7296E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>19</td>
<td>2.7226E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>20</td>
<td>2.7116E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>21</td>
<td>2.7006E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>22</td>
<td>2.6906E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>23</td>
<td>2.6811E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>24</td>
<td>2.6722E-03</td>
<td>2.8079E-03</td>
</tr>
<tr>
<td>25</td>
<td>2.6642E-03</td>
<td>2.8079E-03</td>
</tr>
</tbody>
</table>

Figure 21b. Convergence History
Figure 21c. Timing Information
<table>
<thead>
<tr>
<th>Index</th>
<th>X</th>
<th>R</th>
<th>S</th>
<th>WACH</th>
<th>PHI</th>
<th>CP</th>
<th>Q</th>
<th>U</th>
<th>U-RADIAL</th>
<th>U-THETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1519</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1520</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1521</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1522</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1523</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1524</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1525</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1526</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1527</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1528</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
<tr>
<td>1529</td>
<td>92.5000</td>
<td>47.1843</td>
<td>156.277</td>
<td>-89.15</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
<td>-90.00</td>
</tr>
</tbody>
</table>

**Figure 21d. Surface Properties – θ-Constant Cuts**
<table>
<thead>
<tr>
<th>R</th>
<th>PACK</th>
<th>PHI</th>
<th>CP</th>
<th>U</th>
<th>U-RADIAL</th>
<th>U-THETA</th>
<th>W</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>.5598</td>
<td>81.3583</td>
<td>-3.0756</td>
<td>.5372</td>
<td>-1.1019</td>
<td>-0.0000</td>
<td>.1019</td>
<td>.0000</td>
</tr>
<tr>
<td>5.000</td>
<td>.5703</td>
<td>80.8256</td>
<td>-3.2107</td>
<td>.5451</td>
<td>-1.1211</td>
<td>-0.0000</td>
<td>.1121</td>
<td>.0000</td>
</tr>
<tr>
<td>10.000</td>
<td>.5688</td>
<td>80.2368</td>
<td>-3.4235</td>
<td>.5578</td>
<td>-1.1247</td>
<td>-0.0000</td>
<td>.1124</td>
<td>.0000</td>
</tr>
<tr>
<td>15.000</td>
<td>.6095</td>
<td>79.5766</td>
<td>-3.7209</td>
<td>.5971</td>
<td>-1.1410</td>
<td>-0.0000</td>
<td>.1410</td>
<td>.0000</td>
</tr>
<tr>
<td>20.000</td>
<td>.6390</td>
<td>78.8273</td>
<td>-4.1142</td>
<td>.5969</td>
<td>-1.1629</td>
<td>-0.0000</td>
<td>.1629</td>
<td>.0000</td>
</tr>
<tr>
<td>25.000</td>
<td>.6662</td>
<td>78.1007</td>
<td>-4.4809</td>
<td>.6159</td>
<td>-1.1845</td>
<td>-0.0000</td>
<td>.1845</td>
<td>.0000</td>
</tr>
<tr>
<td>27.500</td>
<td>.6982</td>
<td>77.4430</td>
<td>-4.9192</td>
<td>.6368</td>
<td>-1.2116</td>
<td>-0.0000</td>
<td>.2116</td>
<td>.0000</td>
</tr>
<tr>
<td>31.250</td>
<td>.7356</td>
<td>76.5920</td>
<td>-5.4352</td>
<td>.6591</td>
<td>-1.2465</td>
<td>-0.0000</td>
<td>.2465</td>
<td>.0000</td>
</tr>
<tr>
<td>35.000</td>
<td>.7800</td>
<td>75.5947</td>
<td>-6.0511</td>
<td>.6822</td>
<td>-1.2908</td>
<td>-0.0000</td>
<td>.2908</td>
<td>.0000</td>
</tr>
<tr>
<td>37.500</td>
<td>.8151</td>
<td>74.8262</td>
<td>-6.3404</td>
<td>.6982</td>
<td>-1.3275</td>
<td>-0.0000</td>
<td>.3275</td>
<td>.0000</td>
</tr>
<tr>
<td>40.000</td>
<td>.8577</td>
<td>73.9574</td>
<td>-7.1304</td>
<td>.7151</td>
<td>-1.3726</td>
<td>-0.0000</td>
<td>.3726</td>
<td>.0000</td>
</tr>
<tr>
<td>42.500</td>
<td>.9177</td>
<td>72.6479</td>
<td>-7.8444</td>
<td>.7336</td>
<td>-1.4311</td>
<td>-0.0000</td>
<td>.4311</td>
<td>.0000</td>
</tr>
<tr>
<td>45.000</td>
<td>.9917</td>
<td>71.4017</td>
<td>-8.4943</td>
<td>.7546</td>
<td>-1.5138</td>
<td>-0.0000</td>
<td>.5138</td>
<td>.0000</td>
</tr>
<tr>
<td>46.250</td>
<td>1.0447</td>
<td>71.1285</td>
<td>-9.6405</td>
<td>.7662</td>
<td>-1.5670</td>
<td>-0.0000</td>
<td>.5670</td>
<td>.0000</td>
</tr>
<tr>
<td>47.500</td>
<td>1.1325</td>
<td>70.3043</td>
<td>-10.7499</td>
<td>.7796</td>
<td>-1.6308</td>
<td>-0.0000</td>
<td>.6308</td>
<td>.0000</td>
</tr>
<tr>
<td>48.750</td>
<td>1.2246</td>
<td>69.5514</td>
<td>-11.8134</td>
<td>.7992</td>
<td>-1.7087</td>
<td>-0.0000</td>
<td>.7087</td>
<td>.0000</td>
</tr>
<tr>
<td>49.1134</td>
<td>1.2676</td>
<td>65.2790</td>
<td>-12.2863</td>
<td>.8136</td>
<td>-1.7589</td>
<td>-0.0000</td>
<td>.7589</td>
<td>.0000</td>
</tr>
</tbody>
</table>

**SEISMIC MASS FLOW**: 909.0689

*Figure 21e. Field Properties*
<table>
<thead>
<tr>
<th>X</th>
<th>MASS FLOW</th>
<th>PER CENT ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>46.750</td>
<td>11666.9384</td>
<td>-.06</td>
</tr>
<tr>
<td>47.500</td>
<td>11667.1487</td>
<td>-.01</td>
</tr>
<tr>
<td>48.250</td>
<td>11667.3590</td>
<td>-.00</td>
</tr>
<tr>
<td>49.000</td>
<td>11667.5693</td>
<td>-.00</td>
</tr>
<tr>
<td>49.750</td>
<td>11667.7796</td>
<td>-.00</td>
</tr>
<tr>
<td>50.500</td>
<td>11667.9899</td>
<td>-.00</td>
</tr>
<tr>
<td>51.250</td>
<td>11668.1902</td>
<td>-.00</td>
</tr>
<tr>
<td>52.000</td>
<td>11668.3905</td>
<td>-.00</td>
</tr>
<tr>
<td>52.750</td>
<td>11668.5908</td>
<td>-.00</td>
</tr>
<tr>
<td>53.500</td>
<td>11668.7911</td>
<td>-.00</td>
</tr>
<tr>
<td>54.250</td>
<td>11668.9914</td>
<td>-.00</td>
</tr>
<tr>
<td>55.000</td>
<td>11669.1917</td>
<td>-.00</td>
</tr>
<tr>
<td>55.750</td>
<td>11669.3920</td>
<td>-.00</td>
</tr>
<tr>
<td>56.500</td>
<td>11669.5923</td>
<td>-.00</td>
</tr>
<tr>
<td>57.250</td>
<td>11669.7926</td>
<td>-.00</td>
</tr>
<tr>
<td>58.000</td>
<td>11670.0929</td>
<td>-.00</td>
</tr>
<tr>
<td>58.750</td>
<td>11670.2932</td>
<td>-.00</td>
</tr>
<tr>
<td>59.500</td>
<td>11670.4935</td>
<td>-.00</td>
</tr>
<tr>
<td>60.250</td>
<td>11670.6938</td>
<td>-.00</td>
</tr>
<tr>
<td>61.000</td>
<td>11670.8941</td>
<td>-.00</td>
</tr>
<tr>
<td>61.750</td>
<td>11671.0944</td>
<td>-.00</td>
</tr>
<tr>
<td>62.500</td>
<td>11671.2947</td>
<td>-.00</td>
</tr>
<tr>
<td>63.250</td>
<td>11671.4950</td>
<td>-.00</td>
</tr>
<tr>
<td>64.000</td>
<td>11671.6953</td>
<td>-.00</td>
</tr>
<tr>
<td>64.750</td>
<td>11671.8956</td>
<td>-.00</td>
</tr>
<tr>
<td>65.500</td>
<td>11672.0959</td>
<td>-.00</td>
</tr>
<tr>
<td>66.250</td>
<td>11672.2962</td>
<td>-.00</td>
</tr>
<tr>
<td>67.000</td>
<td>11672.4965</td>
<td>-.00</td>
</tr>
<tr>
<td>67.750</td>
<td>11672.6968</td>
<td>-.00</td>
</tr>
<tr>
<td>68.500</td>
<td>11672.8971</td>
<td>-.00</td>
</tr>
<tr>
<td>69.250</td>
<td>11673.0974</td>
<td>-.00</td>
</tr>
<tr>
<td>70.000</td>
<td>11673.2977</td>
<td>-.00</td>
</tr>
<tr>
<td>70.750</td>
<td>11673.4980</td>
<td>-.00</td>
</tr>
<tr>
<td>71.500</td>
<td>11673.6983</td>
<td>-.00</td>
</tr>
<tr>
<td>72.250</td>
<td>11673.8986</td>
<td>-.00</td>
</tr>
<tr>
<td>73.000</td>
<td>11674.0989</td>
<td>-.00</td>
</tr>
<tr>
<td>73.750</td>
<td>11674.2992</td>
<td>-.00</td>
</tr>
<tr>
<td>74.500</td>
<td>11674.4995</td>
<td>-.00</td>
</tr>
<tr>
<td>75.250</td>
<td>11674.6998</td>
<td>-.00</td>
</tr>
<tr>
<td>76.000</td>
<td>11674.8991</td>
<td>-.00</td>
</tr>
<tr>
<td>76.750</td>
<td>11675.0994</td>
<td>-.00</td>
</tr>
<tr>
<td>77.500</td>
<td>11675.2997</td>
<td>-.00</td>
</tr>
<tr>
<td>78.250</td>
<td>11675.4998</td>
<td>-.00</td>
</tr>
<tr>
<td>79.000</td>
<td>11675.6999</td>
<td>-.00</td>
</tr>
<tr>
<td>79.750</td>
<td>11675.8996</td>
<td>-.00</td>
</tr>
<tr>
<td>80.500</td>
<td>11676.0999</td>
<td>-.00</td>
</tr>
<tr>
<td>81.250</td>
<td>11676.2995</td>
<td>-.00</td>
</tr>
<tr>
<td>82.000</td>
<td>11676.5999</td>
<td>-.00</td>
</tr>
<tr>
<td>82.750</td>
<td>11676.7997</td>
<td>-.00</td>
</tr>
<tr>
<td>83.500</td>
<td>11677.0995</td>
<td>-.00</td>
</tr>
<tr>
<td>84.250</td>
<td>11677.2992</td>
<td>-.00</td>
</tr>
<tr>
<td>85.000</td>
<td>11677.4989</td>
<td>-.00</td>
</tr>
<tr>
<td>85.750</td>
<td>11677.6987</td>
<td>-.00</td>
</tr>
<tr>
<td>86.500</td>
<td>11677.8984</td>
<td>-.00</td>
</tr>
<tr>
<td>87.250</td>
<td>11678.0980</td>
<td>-.00</td>
</tr>
<tr>
<td>88.000</td>
<td>11678.2977</td>
<td>-.00</td>
</tr>
<tr>
<td>88.750</td>
<td>11678.4973</td>
<td>-.00</td>
</tr>
<tr>
<td>89.500</td>
<td>11678.6969</td>
<td>-.00</td>
</tr>
<tr>
<td>90.250</td>
<td>11678.8965</td>
<td>-.00</td>
</tr>
<tr>
<td>91.000</td>
<td>11679.0958</td>
<td>-.00</td>
</tr>
<tr>
<td>91.750</td>
<td>11679.2952</td>
<td>-.00</td>
</tr>
<tr>
<td>92.500</td>
<td>11679.4946</td>
<td>-.00</td>
</tr>
<tr>
<td>93.250</td>
<td>11679.6940</td>
<td>-.00</td>
</tr>
<tr>
<td>94.000</td>
<td>11679.8934</td>
<td>-.00</td>
</tr>
<tr>
<td>94.750</td>
<td>11680.0928</td>
<td>-.00</td>
</tr>
<tr>
<td>95.500</td>
<td>11680.2921</td>
<td>-.00</td>
</tr>
<tr>
<td>96.250</td>
<td>11680.4915</td>
<td>-.00</td>
</tr>
<tr>
<td>97.000</td>
<td>11680.6909</td>
<td>-.00</td>
</tr>
<tr>
<td>97.750</td>
<td>11680.8902</td>
<td>-.00</td>
</tr>
</tbody>
</table>

*Figure 21f. Mass-Flow Conservation*
### SURFACE FLOW PROPERTIES - \( X = \) CONSTANT CUTS

<table>
<thead>
<tr>
<th>INDEX</th>
<th>( X )</th>
<th>( R )</th>
<th>( \Theta )</th>
<th>( S )</th>
<th>( MACH )</th>
<th>( CP )</th>
<th>( PHI )</th>
<th>( PHI_s )</th>
<th>( U )</th>
<th>( U)-RADIAL</th>
<th>( U)-( \Theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>40.0000</td>
<td>39.5966</td>
<td>0.0000</td>
<td>0.0000</td>
<td>5784</td>
<td>-3.3143</td>
<td>78.1219</td>
<td>56.39</td>
<td>0.0000</td>
<td>4860</td>
<td>-2859</td>
</tr>
<tr>
<td>48</td>
<td>40.0000</td>
<td>36.6364</td>
<td>33.7350</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>41</td>
<td>40.0000</td>
<td>36.6466</td>
<td>33.7550</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>39</td>
<td>40.0000</td>
<td>36.6566</td>
<td>33.7750</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>40</td>
<td>40.0000</td>
<td>36.6666</td>
<td>33.7950</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>45</td>
<td>40.0000</td>
<td>36.6766</td>
<td>33.8150</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>46</td>
<td>40.0000</td>
<td>36.6866</td>
<td>33.8350</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>47</td>
<td>40.0000</td>
<td>36.6966</td>
<td>33.8550</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
<tr>
<td>48</td>
<td>40.0000</td>
<td>36.7066</td>
<td>33.8750</td>
<td>15.0975</td>
<td>8120</td>
<td>-6.6789</td>
<td>81.4317</td>
<td>7666</td>
<td>-0.0644</td>
<td>7651</td>
<td>-0.0583</td>
</tr>
</tbody>
</table>

---

Figure 21g. Surface Properties – All Cuts
Figure 21g. Concluded
Table 4. Headings for Surface Point and Surface Geometry Printout

<table>
<thead>
<tr>
<th>INDEX</th>
<th>Surface point index</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>R</td>
<td>r</td>
</tr>
<tr>
<td>THETA</td>
<td>θ</td>
</tr>
<tr>
<td>NX</td>
<td>$n_x$, component of unit surface normal</td>
</tr>
<tr>
<td>NR</td>
<td>$n_r$, component of unit surface normal</td>
</tr>
<tr>
<td>NT</td>
<td>$n_\theta$, component of unit surface normal</td>
</tr>
<tr>
<td>S</td>
<td>s, arc length along cut</td>
</tr>
<tr>
<td>PHI.S-XC</td>
<td>$\phi_{s,x}$, velocity along x constant cut</td>
</tr>
<tr>
<td>PHI.S-RC</td>
<td>$\phi_{s,r}$, velocity along r constant cut</td>
</tr>
<tr>
<td>PHI.S-TC</td>
<td>$\phi_{s,\theta}$, velocity along $\theta$ constant cut</td>
</tr>
<tr>
<td>I</td>
<td>i, j, k of field point adjacent to surface point</td>
</tr>
<tr>
<td>J</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
</tr>
<tr>
<td>SURFARM</td>
<td>Δx, Δr or Δθ between surface point and adjacent field point</td>
</tr>
<tr>
<td>TYPE</td>
<td></td>
</tr>
<tr>
<td>-1 = x intersect</td>
<td></td>
</tr>
<tr>
<td>0 = theta intersect</td>
<td></td>
</tr>
<tr>
<td>1 = r intersect</td>
<td></td>
</tr>
<tr>
<td>2 = both mesh and surface node</td>
<td></td>
</tr>
</tbody>
</table>
Table 5. Convergence History Headings

**SWEEP**

Relaxation sweep number, (n)

**FIELD POINTS**

| AVE RESIDUE | Sum of \(|\psi_{i,j,k}^{(n)}-\psi_{i,j,k}^{(n-1)}| \) over all points i,j,k in the flowfield divided by \(\phi_{\text{max}}^{(n)}-\phi_{\text{min}}^{(n)}\) and the number of such field points |
| I, J, K | The indices i,j,k of the field point having the maximum change in \(\phi\) and the maximum value of \(|\psi_{i,j,k}^{(n)}-\psi_{i,j,k}^{(n-1)}| \) divided by \(\phi_{\text{max}}^{(n)}-\phi_{\text{min}}^{(n)}\) |

"CONVERGING/ DIVERGING"

**SURFACE POINTS**

| AVE RESIDUE | Sum of \(|\phi_{S}^{(n)}-\phi_{S}^{(n-1)}| \) over all surface points S divided by \(\phi_{\text{max}}^{(n)}-\phi_{\text{min}}^{(n)}\) and the number of surface points |
| I, J, K, INDEX | The indices i,j,k of the field point next to the surface point with the maximum change in \(\phi\), the index of that surface point, and the maximum value of \(|\phi_{S}^{(n)}-\phi_{S}^{(n-1)}| \) divided by \(\phi_{\text{max}}^{(n)}-\phi_{\text{min}}^{(n)}\). |

M > 1.0

number of field points for which Mach number > 1.0

EIGEN1

1/(1-\(\lambda_1\)), equations 51,53

EIGEN2

1/(1-\(\lambda_2\)), equations 51,54

"EXTRAPOLATION FLAG" * indicates flowfield extrapolation using equation (52) was made after this sweep
Table 6. Surface-Point Printout Headings

<table>
<thead>
<tr>
<th>INDEX</th>
<th>Surface point number</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>x</td>
</tr>
<tr>
<td>R</td>
<td>r</td>
</tr>
<tr>
<td>THETA</td>
<td>θ</td>
</tr>
<tr>
<td>S</td>
<td>s, arc length along the cut of the surface</td>
</tr>
<tr>
<td>MACH</td>
<td>Mach Number</td>
</tr>
<tr>
<td>CP</td>
<td>$C_p$, coefficient of pressure, $(p/p_\infty-1)\gamma M^2$ (if $M_\infty = 0.0$, then $C_p = (p/p_\infty-1)\gamma$, where the * indicates sonic conditions)</td>
</tr>
<tr>
<td>PHI</td>
<td>φ</td>
</tr>
<tr>
<td>Q</td>
<td>$q = (u^2 + u_r^2 + u_\theta^2)^{1/2}$</td>
</tr>
<tr>
<td>PHI,S</td>
<td>$u_s$, component of velocity along the cut</td>
</tr>
<tr>
<td>U</td>
<td>u</td>
</tr>
<tr>
<td>U-RADIAL</td>
<td>$u_r$</td>
</tr>
<tr>
<td>U-THETA</td>
<td>$u_\theta$</td>
</tr>
</tbody>
</table>
Table 7. Field-Point Printout Headings

<table>
<thead>
<tr>
<th>R</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACH</td>
<td>M</td>
</tr>
<tr>
<td>PHI</td>
<td>φ</td>
</tr>
<tr>
<td>CP</td>
<td>$C_p$, coefficient of pressure, $(p/p_{∞} - 1)^{1/2} \gamma M^2$ if $M_{∞} = 0.0$, then $C_p = (p/p_{*} - 1)^{1/2} \gamma$, where the * indicates sonic conditions</td>
</tr>
<tr>
<td>U</td>
<td>u</td>
</tr>
<tr>
<td>U-RADIAL</td>
<td>$u_r$</td>
</tr>
<tr>
<td>U-THETA</td>
<td>$u_θ$</td>
</tr>
<tr>
<td>V</td>
<td>v</td>
</tr>
<tr>
<td>W</td>
<td>w</td>
</tr>
</tbody>
</table>

3.4 Run Quality

A major concern of the user is the degree to which the computed solution matches a real flowfield. This can be a relatively difficult question to answer. The primary effects are the nature of the flow analyzed, the accuracy of the analysis if a fully-converged solution is obtained, and the degree of convergence. The program predicts potential flow, which is inviscid irrotational flow. A necessary, but not sufficient, condition for acceptable flow prediction is that the flow to be analyzed not be greatly effected by viscous and rotational effects.

If the flow is one for which potential flow is a good approximation, the next question is the acceptability of the mesh placement. If the flow is a duct, the mass-flow integration will give a check on accuracy. Otherwise, the best way to estimate accuracy (assuming a converged solution) is by comparison with experiments for the same or similar configurations.

The question of convergence can be answered, to a large extent, by looking at the AVE RESIDUE in the convergence history. Values of $10^{-6}$ are usually satisfactory. For any given class of problems a few extra long runs can be made to obtain $10^{-6}$ or better convergence, and the results compared with lesser levels to see if there are any significant changes in the answers.
3.5 DIAGNOSTICS AND TROUBLESHOOTING

Failure of the program to successfully provide a flow analysis for a data case could be due to a number of causes. Among these are input errors, errors in the program, and geometries the program was not written to handle. As obtaining help to determine if the program is in error can be very time consuming, it is preferable for a user to first check for input errors. The program does a considerable amount of input checking and many of the diagnostics are self-explanatory. The first step in searching for the cause of any run failure should be to check the printout for any diagnostic messages.

One task of the program is checking the input geometry for completeness and self-consistency. Many of the error messages in this section are self-explanatory. One area that may be confusing is the messages from the subroutine which generates internal maps. There is a subroutine in the program that generates the maps that essentially list the surface points in sequence along \(x\), \(r\), and \(\theta\) constant cuts. If there are points missing, or points that are not consistent with each other, this subroutine will print diagnostic messages. The code works by starting at a surface point on a cut and then trying to follow the cut by finding the surface points in sequence on the cut. If the subroutine cannot find the next point on a cut, and the previous point was not a boundary point of the computational flowfield, an error message is printed and the points on the cut that have already been found will be printed. The problem area is at the end of the printed cut. Some more information may be obtained by looking at similar error printouts next to the one of interest as, usually, the parts of a cut on either side of a bad or missing surface point will be printed separately. The procedure for correcting the input error is to examine the area of difficulty, usually by sketches or plots, to determine the nature of the problem. If all surface points near the problem area are plotted versus the mesh, the problem is usually obvious.

In interpreting the points printed out, the user should be aware that the mapping subroutine starts at a point and tracks the curve until it ends on a boundary or closes on itself. If the cut ends on a boundary, it then is followed in the opposite direction from the start point to the other end of the cut. If the subroutine is successful, the two segments of the curve are connected together in sequence. If the mapping routine fails while following the second segment, the points printed will be in sequence from the start point to the boundary point, followed by a sequence in the opposite sense from the point next to the start point to the problem area.

The most common geometry problems are illustrated in figure 22. The problem of figure 22a is simply a point missing from the geometry file. The problem shown in figure 22b is inconsistent geometry points. Moving A to A' or B to B' would make the points consistent. This condition is usually caused by a tolerance problem in the geometry/mesh intersection program. Points A and B are very close to the mesh intersection and have been found by separate passes of the geometry processing code. A small error in position is enough to put the point on the wrong side of the mesh intersect. Often this problem can be fixed by replacing points A and B with a single point at the mesh intersect. Remember there is usually a point C in the third coordinate which should also be consolidated with A and B at the mesh intersect.
a.) MISSING SURFACE POINT

b.) INCONSISTENT SURFACE POINTS

Figure 22. Typical Geometry Problems
REFERENCES


APPENDIX A

TRIDIAGONAL EQUATION SOLVER

The following is the logic for direct solution of a system of N simultaneous linear equations which are of tridiagonal form. If the unknown is \( \phi_j, j=1,2...N \), the equations are tridiagonal if they can be ordered such that the jth equations involves only \( \phi_{j-1}, \phi_j \) and \( \phi_{j+1} \).

The difference equations are:

\[
Y_1 \phi_1 + Z_1 \phi_2 = G_1 \quad \text{(A-1)}
\]
\[
X_j \phi_{j-1} + Y_j \phi_j + Z_j \phi_{j+1} = G_j \quad 2 \leq j \leq N-1 \quad \text{(A-2)}
\]
\[
X_N \phi_{N-1} + Y_N \phi_N = G_N \quad \text{(A-3)}
\]

For the above notation \( \phi_j \) represents \( \phi_{i,j,k} \) as the lines for SLOR in this program are radial lines for which \( j \) is the radial index.

The equations can be written in matrix notation as

\[
PW = R \quad \text{(A-4)}
\]

where

\[
P = \begin{bmatrix}
Y_1 & Z_1 & & & \\
X_2 & Y_2 & Z_2 & & \\
& X_3 & Y_3 & Z_3 & \\
& & & \ddots & \ddots \\
& & & X_{N-2} & Y_{N-2} \end{bmatrix} \quad \text{(A-5)}
\]

\[
W = \begin{bmatrix}
\phi_1 \\
\phi_2 \\
\phi_3 \\
\vdots \\
\phi_N
\end{bmatrix} \quad \text{(A-6)}
\]

\[
R = \begin{bmatrix}
G_1 \\
G_2 \\
G_3 \\
\vdots \\
G_N
\end{bmatrix} \quad \text{(A-7)}
\]
P can be decomposed into two triangular matrices, M and N, such that

\[ P = MN \]  \hspace{1cm} (A-8)

where

\[
M = \begin{bmatrix}
M_1 \\
X_2 & M_2 \\
X_3 & M_3 \\
& \ddots \\
& & \ddots & X_{N-2} & M_{N-2} \\
& & & \ddots & X_{N-1} & M_{N-1} \\
& & & & \ddots & X_N & M_N
\end{bmatrix}
\]  \hspace{1cm} (A-9)

and

\[
N = \begin{bmatrix}
I & N_1 \\
I & N_2 \\
I & N_3 \\
& \ddots \\
& & \ddots & I & N_{N-2} \\
& & & \ddots & I & N_{N-1} \\
& & & & \ddots & I & I
\end{bmatrix}
\]  \hspace{1cm} (A-10)

The following relations are then obtained:

\[
M_1 = Y_1
\]
\[
M_j = Y_j - X_jN_{j-1} \hspace{1cm} 2 \leq j \leq N
\]  \hspace{1cm} (A-11)
\[
N_j = Z_j/M_j \hspace{1cm} 1 \leq j \leq N-1
\]
The matrix $U$ is defined by

$$MU = R \tag{A-12}$$

where

$$U = \begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
\vdots \\
U_N
\end{bmatrix} \tag{A-13}$$

which gives the relations

$$U_1 = G_1/M_1 \tag{A-14}$$

$$U_j = (G_j - X_j U_{j-1})/M_j, \quad 2 \leq j \leq N$$

When equations (A-8) and (A-12) are substituted into equation (A-4) and the resulting equation is multiplied by $M^{-1}$, the following matrix equation is obtained:

$$NW = U \tag{A-15}$$

which leads to the following equations for the $\phi_j$:

$$\phi_N = U_N \tag{A-16}$$

$$\phi_j = U_j - N_j \phi_{j+1}, \quad 1 \leq j \leq N-1$$

The solution procedure to obtain the $\phi_j$ is to use (A-11) to obtain $M_j$ and $N_j$ and (A-14) to obtain the $U_j$ for all $j$. Then (A-16) is used to calculate the $\phi_j$. 

---

80
APPENDIX B

PARAMETRIC CUBIC INTERPOLATION

Parametric cubic interpolation is used for interpolating to finer meshes as well as in several other areas of the program. It was chosen because odd-order interpolation leads to symmetric formulas, which are preferrable, and first-order or linear interpolation was thought to be too inaccurate. Also, cubic interpolation, as used with function and first derivatives, requires information at only two points. Given points A and B and a parameter u which is zero at A and 1 at B, then a function f(u) can be expressed as

\[ f(u) = C_0 + C_1u + C_2u^2 + C_3u^3 \]  

(B-1)

however, the form

\[ f(u) = F_0(u)f(0) + F_1(u)f(1) + D_0(u)f_u(0) + D_1(u)f_u(1) \]

(B-2)

where the \( F_i \) and \( D_i \) are cubics in \( u \), is preferrable for use in the code as it is easier to evaluate.

It can be shown with a little algebra that

\[ F_0(u) = (1-u)^2(1+2u) \]

\[ F_1(u) = u^2(3-2u) \]

\[ D_0(u) = u(1-u)^2 \]

\[ D_1(u) = -u^2(1-u) \]  

(B-3)

The formulas are expressed in parametric form for simplicity. If \( x \) is the coordinate of interest and the known points are \( x_0 \) and \( x_1 \):

\[ u = \frac{x-x_0}{x_1-x_0} \]  

(B-4)

\[ f_u = \frac{x_1-x_0}{f_x} \]  

(B-5)
The general interpolation formula for a unit cubic box with parametric coordinates \( u, v, \) and \( w \) is

\[
f(u,v,w) = \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} \left[ F_i(u)F_j(v)F_k(w)f_{i,j,k} + D_i(u)F_j(v)F_k(w)f_{ui,j,k} + F_i(u)D_j(v)F_k(w)f_{vi,j,k} + F_i(u)F_j(v)D_k(w)f_{wi,j,k} + D_i(u)D_j(v)F_k(w)f_{uv,i,j,k} + F_i(u)D_j(v)D_k(w)f_{vw,i,j,k} + D_i(u)F_j(v)D_k(w)f_{wu,i,j,k} + D_i(u)D_j(v)D_k(w)f_{uvw,i,j,k} \right]
\]

(B-6)

where

\[ f_{i,j,k} = f(u_i,v_j,w_k) \]

and

\[ u_0 = 0, u_1 = 1.0, v_0 = 0, v_1 = 1.0, w_0 = 0, w_1 = 1.0. \]

The above formula is used in the code for interpolation, except that the last four lines of equation B-6 are neglected. While this is incorrect, and introduces errors of unknown magnitude, it greatly simplifies the coding. Errors in the interpolation due to neglect of terms do not cause errors in the final flow solution, but instead increase the cost of a solution by requiring additional relaxation sweeps to eliminate the errors due to the interpolation process.

Referring to figure B-1 for notation, the value of a function \( f \) at \((u,v,w)\) can either be calculated from equation B-6 or by the following procedure:

![Figure B-1 Rectangular Box for Interpolation](image)
First calculate,

\[
f_A = f(0,v,0) = \sum_{j=0}^{1} \left[ F_j(v)f_{0,j,0} + D_j(v)f_{v0,j,0} \right]
\]
\[
f_{uA} = f_u(0,v,0) = \sum_{j=0}^{1} \left[ F_j(v)f_{u0,j,0} + D_j(v)f_{uv0,j,0} \right]
\]
\[
f_{wA} = f_w(0,v,0) = \sum_{j=0}^{1} \left[ F_j(v)f_{w0,j,0} + D_j(v)f_{vw0,j,0} \right]
\]
\[
f_{wuA} = f_{wu}(0,v,0) = \sum_{j=0}^{1} \left[ F_j(v)f_{wu0,j,0} + D_j(v)f_{uw0,j,0} \right]
\]

(B-7)

The same quantities are calculated at A', A'', and A''' by similar formulas. Then calculate

\[
f_B = f(u,v,0) = F_0(u)f_A + F_1(u)f_A' + D_0(u)f_{uA} + D_1(u)f_{uA'}
\]
\[
f_{wB} = f_w(u,v,0) = F_0(u)f_{wA} + F_1(u)f_{wA'} + D_0(u)f_{wuA} + D_1(u)f_{wuA'}
\]
\[
f_B' \text{ and } f_{wB}' \text{ are calculated similarly}
\]

Then

\[
f_C = f(u,v,w) = F_0(w)f_B + F_1(w)f_B' + D_0(w)f_{wB} + D_1(w)f_{wB'}
\]

(B-9)

The above procedure is exactly equivalent to using equation B-6. The advantage of the above procedure is that it can easily be used with minor modifications if one or more corners of the box are missing, which can occur in the vicinity of surfaces. This gives a workable interpolation procedure near surfaces, however, the results, when a corner of the box is missing, are not unique. That is, interpolating first in v, then in u, and finally in w as described, is not equivalent to another sequence. It is equivalent only if the rectangular box is complete.
An analysis has been developed and a computer code, P465 Version A, written for the prediction of transonic potential flow about three-dimensional objects including inlet, duct, and body geometries. Finite differences and line relaxation are used to solve the complete potential flow equation. The coordinate system used for the calculations is independent of body geometry. Cylindrical coordinates have been used for the computer code. The analysis has been programmed in extended FORTRAN IV for the Control Data Corporation CYBER 203 vector computer. The programming of the analysis has been oriented toward taking advantage of the vector processing capabilities of this computer. Comparisons of computed results with experimental measurements are presented to verify the analysis. Descriptions of program input and output formats are provided.