DONBOL: A Computer Program for Predicting Axisymmetric Nozzle Afterbody Pressure Distributions and Drag at Subsonic Speeds

Lawrence E. Putnam

FOR EARLY DOMESTIC DISSEMINATION

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.

Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Date for general release: May 1981
Note that this document bears the label "FEDD," an acronym for "FOR EARLY DOMESTIC DISSEMINATION." The FEDD label is affixed to documents that may contain information having high commercial potential.

The FEDD concept was developed as a result of the desire to maintain U.S. leadership in world trade markets and encourage a favorable balance of trade. Since the availability of tax-supported U.S. technology to foreign business interests could represent an unearned benefit, research results that may have high commercial potential are being distributed to U.S. industry in advance of general release.

The recipient of this report must treat the information it contains according to the conditions of the FEDD label on the front cover.
DONBOL: A Computer Program for Predicting Axisymmetric Nozzle Afterbody Pressure Distributions and Drag at Subsonic Speeds

Lawrence E. Putnam
Langley Research Center
Hampton, Virginia

NASA Technical Memorandum 78779

National Aeronautics and Space Administration
Scientific and Technical Information Office
1979
A Neumann solution for inviscid external flow has been coupled to a modified Reshotko-Tucker integral boundary-layer technique, the control volume method of Presz for calculating flow in the separated region, and an inviscid one-dimensional solution for the jet exhaust flow in order to predict axisymmetric nozzle afterbody pressure distributions and drag. The viscous and inviscid flows are solved iteratively until convergence is obtained. A computer algorithm of this procedure has been written and is called DONBOL. This paper provides a description of the computer program and a guide to its use. Comparisons of the predictions of this method with experiment show that the method accurately predicts the pressure distributions of boattail afterbodies which have the jet exhaust flow simulated by solid bodies. For nozzle configurations which have the jet exhaust simulated by high-pressure air, the present method significantly underpredicts the magnitude of nozzle pressure drag. This deficiency results because the method neglects the effects of jet plume entrainment. This method is limited to subsonic free-stream Mach numbers below that for which the flow over the body of revolution becomes sonic.

The drag-producing components of the airplane propulsion system are usually installed in areas where the flow field is extremely complex. High body slopes and long boundary-layer runs, especially in the afterbody nozzle region, result in strong viscous effects on boattail drag. Furthermore, the viscous nature of the jet exhaust plume complicates the flow in this region. Because of these strong viscous interactions, current methods used for predicting the installed propulsion system drag are usually limited to empirical techniques. Recently, however, investigators have achieved some success in predicting uninstalled drag of axisymmetric nozzles with what is usually called the patched viscous-inviscid technique. (See refs. 1 to 7, for example.) In reference 1, Reubush and Putnam combine iteratively a conventional boundary-layer technique with a linearized potential-flow computation to account for the viscous-inviscid interaction. For boattail nozzles on which boundary-layer separation occurs, Reubush and Putnam employ the discriminating streamline concept of Presz (refs. 8 and 9) to separate the reverse flow region from the outer flow. The patched viscous-inviscid interaction methods have been successful in predicting the qualitative trends in boattail pressure drag with Mach number, Reynolds number, and nozzle geometry in spite of the complexity of the flow even for isolated boattails. (See ref. 1, for example.) In general, however, these techniques substantially underpredict the absolute levels of pressure drag on boattail nozzles at subsonic speeds.

Recently, an improved analytical model of the flow in the separated region has been developed by Presz (refs. 10 and 11). With this analytical model, the effects of axial-pressure gradients, surface skin friction, and jet plume entrainment on the shape of the discriminating streamline are computed. Predictions made using this new technique (refs. 10 and 11) are in substantially
better agreement with experiment than the predictions of the previous methods (refs. 1 to 7). This improved model of the separated flow region, therefore, has been combined iteratively with the inviscid linearized potential-flow solution described in reference 1.

The present paper describes the various components of the resulting computer algorithm called DONBOL. Also, this paper illustrates the prediction capabilities of the method by comparison with experimental data. A user's guide to the computer program is presented. The computer program may be obtained from COSMIC, Suite 112, Barrow Hall, University of Georgia, Athens, GA 30602.

SYMBOLS

The symbols used in the computer printouts are given in a separate column.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SREF</td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>C_d</td>
<td></td>
</tr>
<tr>
<td>C_d_f</td>
<td></td>
</tr>
<tr>
<td>C_d_p</td>
<td></td>
</tr>
<tr>
<td>C_d_t</td>
<td></td>
</tr>
<tr>
<td>C_p</td>
<td>CP</td>
</tr>
<tr>
<td>c_f</td>
<td>CF</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>d_b</td>
<td>DB</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>l</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>MO</td>
</tr>
<tr>
<td>NPR</td>
<td></td>
</tr>
<tr>
<td>N_Re</td>
<td></td>
</tr>
</tbody>
</table>
\( p \)  
static pressure

\( p_t \)  
PT  
total pressure

\( q_{\infty} \)  
free-stream dynamic pressure

\( R \)  
gas constant

\( R_C \)  
body radius corrected for \( \delta^* \) and discriminating streamline

\( r \)  
radial coordinate of cylindrical coordinate system with origin at nose of body of revolution

\( R_{DS} \)  
radius of the discriminating streamline

\( r^* \)  
radius of stream tube for Mach number of 1

\( T_t \)  
TT  
total temperature

\( V_r \)  
VR  
ratio of radial velocity to free-stream velocity

\( V_T \)  
VT  
ratio of local velocity to free-stream velocity

\( V_X \)  
VX  
ratio of axial velocity to free-stream velocity

\( x \)  
X  
axial coordinate of cylindrical coordinate system with origin at nose of body of revolution

\( \Delta x \)  
axial distance downstream of start of boattail

\( \beta \)  
\( = \sqrt{1 - M_{\infty}^2} \)

\( \gamma \)  
ratio of specific heats

\( \delta^* \)  
\( \delta*_L \)  
boundary-layer thickness

\( \eta \)  
boundary-layer displacement thickness

\( \eta_{\infty} \)  
boundary-layer momentum thickness

Subscripts:

\( a \)  
analogous configuration (see eqs. (1) to (6))

\( \text{des} \)  
design conditions of nozzle

\( e \)  
exit

\( \text{exp} \)  
experiment
DESCRIPTION OF METHOD

The present analytical method has been developed to calculate the flow over axisymmetric boattail bodies at subsonic speeds. It is assumed that the flow is composed of a viscous layer near the body, an inviscid external flow, and, if present, an inviscid jet exhaust flow. (See fig. 1.) The effect of the viscous layer is accounted for by modifying the body shape with an appropriate displacement thickness. In the framework of this representation, any boundary-layer separation on the boattail or nozzle surface is accounted for by modifying the afterbody geometry and plume boundary.

Inviscid External Flow Solution

The Neumann solution of reference 12 for incompressible flow over bodies of revolution was used to calculate the inviscid external flow. Since this is a solution for incompressible flow, the compressibility correction of reference 13 was used to correct for Mach number effects. The incompressible flow field considered is that for an "analogous" configuration obtained by means of the affine coordinate transformation given by the following equations:

\[ x_a = \frac{x}{\beta} \]  \hspace{1cm} (1)

\[ r_a = r \]  \hspace{1cm} (2)

where

\[ \beta = \sqrt{1 - M_\infty^2} \]  \hspace{1cm} (3)

The calculated flow velocities of the analogous configuration are then corrected using the following equations:

\[ V_x = \frac{V_{x,a}}{\beta^2} \]  \hspace{1cm} (4)

\[ V_r = \frac{\beta V_{r,a}}{\beta^2} \]  \hspace{1cm} (5)
where

\[ B = \sqrt{1 - M_\infty^2(1 + V_x, a)} \]  

(6)

The pressure coefficients are obtained from the corrected velocities by using the compressible Bernoulli equation and the isentropic flow relations. Experience to date indicates that this compressibility correction provides better agreement with experimental results for flow over boattails than the classic Goethert compressibility correction.

Because the inviscid outer flow solution is based on incompressible flow theory with a compressibility correction, the present method is limited to free-stream Mach numbers for which the flow is subsonic everywhere.

Inviscid Jet Exhaust Flow

To calculate the inviscid boundary of the jet exhaust flow, a procedure based on one-dimensional isentropic flow theory has been developed and is used in the present computer program, DONBOL. The procedure for calculating the radius of the inviscid jet plume at any axial location downstream of the nozzle exit is as follows. Initially, a shape for the jet plume boundary is assumed. Next, the pressure distribution along this boundary is calculated. Then, a new value of the radius at each axial location is ascertained by calculating the cross-sectional area required to expand isentropically from the flow conditions at the nozzle exit to the pressure on the boundary at that location. This new boundary is used in the next iteration as the guess. The equations used to compute the inviscid jet plume boundary from the flow conditions at the nozzle exit and the pressure distribution along the boundary are as follows:

\[
\left( \frac{P_t}{P_{des}} \right) = \left( 1 + \frac{\gamma - 1}{2} M_{des}^2 \right)^{\gamma/(\gamma-1)}
\]

(7)

\[
P_e = \rho_\infty c_p e + p_\infty
\]

(8)

If

\[
\frac{P_{t, jet}}{P_e} > \left( \frac{P_t}{P_{des}} \right)
\]

then

\[ M_{jet} = M_{des} \]

(9)
If

\[
\frac{p_{t, \text{jet}}}{p_e} \leq \left( \frac{p_t}{p} \right)_{\text{des}}
\]

then the static pressure across the exit is assumed equal to the external static pressure at the exit and

\[
M_{\text{jet}} = \sqrt{\frac{2}{\gamma - 1} \left[ \left( \frac{p_{t, \text{jet}}}{p_e} \right)^{(\gamma - 1)/\gamma} - 1 \right]}
\]  

(10)

Then

\[
\frac{r^*}{r_e} = \sqrt{M_{\text{jet}} \left( \frac{\gamma + 1}{2} \right)^{(\gamma + 1)/2(\gamma - 1)} \left( 1 + \frac{\gamma - 1}{2} M_{\text{jet}}^2 \right)^{-(\gamma + 1)/2(\gamma - 1)}}
\]

(11)

Now at any given x-location downstream of nozzle exit since \( C_p \) is a function of \( x \),

\[
p = q_\infty C_p + p_\infty
\]

(12)

\[
M = \sqrt{\frac{2}{\gamma - 1} \left[ \left( \frac{p_{t, \text{jet}}}{p} \right)^{(\gamma - 1)/\gamma} - 1 \right]}
\]

(13)

and

\[
\frac{r^*}{r} = \sqrt{M \left( \frac{\gamma + 1}{2} \right)^{(\gamma + 1)/2(\gamma - 1)} \left( 1 + \frac{\gamma - 1}{2} M^2 \right)^{-(\gamma + 1)/2(\gamma - 1)}}
\]

(14)

Then

\[
\frac{r}{r_e} = \frac{r^*/r_e}{r^*/r}
\]

(15)
Also

\[ V_{x,\text{jet}} = \frac{M}{\sqrt{\frac{YRT_{t,\text{jet}}}{1 + \frac{Y - 1}{2} M^2}}} \quad (16) \]

This procedure has been used to calculate the shape of the inviscid jet plume for the exhaust flow from a convergent nozzle at two nozzle pressure ratios. The procedure is compared in figure 2 with the predictions of the method of Salas (ref. 14) modified to account for pressure variation along the jet boundary. The identical longitudinal pressure distribution along the boundary of the jet was assumed with each method. However, slightly different pressure distributions were assumed for each nozzle pressure ratio. At NPR = 2.90, the shapes of the jet exhaust plume boundary predicted by the two methods are in very good agreement. At NPR = 5.03, the one-dimensional method does not agree as well with the method of Salas. As will be shown later, however, the one-dimensional method does provide a reasonable estimate of the effects of NPR on nozzle drag.

**Viscous Flow**

The properties of the viscous boundary layer (both attached and separated) and the location of any separation on the nozzle boattail are calculated using the methods and computer algorithm developed by Presz, King, and Buteau and described in reference 10. Presz, King, and Buteau computed the turbulent boundary-layer displacement-thickness distribution along the body with the method described in reference 15. This method is a modified version of the Reshotko-Tucker integral boundary-layer solution (ref. 16). A comparison of the predictions of this technique with the experimental measurements of Winter, Rotta, and Smith (ref. 17) at \( \text{M}_\infty = 0.6 \) and a Reynolds number, based on body length, of \( 9.85 \times 10^6 \) is presented in figure 3.

If boundary-layer separation occurs on the boattail, the boundary-layer equations become singular at the separation point. To overcome this difficulty, Presz uses the concept of a discriminating streamline to separate the reverse flow region from the outer boundary-layer flow. This method, described in reference 10, accounts for the effects of axial-pressure gradients, surface skin friction, and viscous mixing in the jet exhaust flow on the shape of this discriminating streamline. Note that the present method does not account for the effects of viscous mixing downstream of the reattachment point.

The use of Presz's model of the separated region requires that some method be available for predicting the location of separation. Several methods are available. They include Presz's control volume technique (ref. 8), Goldschmied's criterion (ref. 18), a modified Page criterion (ref. 19), and Stratford's criterion (ref. 20). A discussion of the accuracy of the various separation location criteria is given in reference 21 by Abeyounis. Any of these methods can be used in the current computer program.
Viscous-Inviscid Interaction

Since the boundary-layer displacement thickness, the discriminating streamline shape, and the inviscid jet boundary are functions of the pressure distribution along the body and the jet boundary, the final converged solution must be obtained by iteration between the inviscid outer flow solution, the inviscid jet plume solution, and the viscous boundary-layer solution. The iteration algorithm used in the present method is shown in figure 4 and is as follows:

(1) Calculate the inviscid pressure distribution on the body of revolution.
(2) Calculate the inviscid jet plume boundary.
(3) Calculate the boundary-layer displacement thickness.
(4) Calculate the location of boundary-layer separation on the boattail. The separation location is calculated using the criteria selected by the user and is based on the pressure distribution and, in some cases, boundary-layer characteristics of the flow over the body. For the first iteration, a separation location will always be predicted. Ideally, the separation location should move aft with increasing number of iterations, and the separation region should essentially disappear as the solution approaches convergence for nozzles and flow conditions where no boundary-layer separation would occur. Unfortunately, with the available separation criteria, this separation region does not always disappear. It is suggested that a solution first be attempted assuming attached flow for nozzles when there is a question about whether or not separation occurs. If the solution diverges, the user can then assume that the flow is not attached, and the calculation must be repeated assuming that the flow is separated.
(5) If a separated flow calculation is required, calculate the shape of the discriminating streamline. To speed convergence and to eliminate some initial numerical stability problems, the present method assumes that for the first four iterations, axial-pressure gradients do not affect the shape of the discriminating streamline. After nine iterations, the shape of the discriminating streamline is frozen.
(6) Correct the body geometry for boundary-layer displacement effects by adding an effective displacement thickness to the original body. The effective displacement thickness includes the discriminating streamline in the separation region. A relaxation procedure described in reference 8 is used to expedite convergence and to eliminate instabilities in the iteration procedure.
(7) Repeat steps (1) to (6) for the desired number of iterations. In the present algorithm, no convergence criteria are specified. Convergence is assumed to occur when two successive iterations plotted to a reasonable scale give essentially the same results. To obtain this result, most configurations require about 15 iterations.
COMPARISONS OF PREDICTIONS AND EXPERIMENT

The predictions of program DONBOL for an \( l/D = 1.768, \ d_b/D = 0.51 \) circular-arc afterbody with a solid cylindrical jet plume simulator and with attached boundary-layer flow are compared with the experimental data of reference 22 in figure 5. At both free-stream Mach numbers shown, the agreement between the predicted and experimental pressure distributions is excellent. The boattail pressure drag of the configuration is underpredicted. However, the differences between theory and experimental drag are within the accuracy of the experimental measurements. These results are typical of all attached-flow cases computed to date with DONBOL.

For boattail nozzles and afterbodies on which the boundary layer separates, the agreement between the predictions of the present method and experiment depends on the chosen separation criterion. In reference 21 Abeyounis showed that a criterion predicts significantly different locations for separation depending on whether the theoretical inviscid pressure distribution or the experimental pressure distribution is used. This result suggests that the predicted separation location may also be a function of the iteration algorithm used in a patched viscous-inviscid interaction procedure such as DONBOL. Therefore, the accuracy of a given separation criterion should be assessed using the total prediction algorithm for which it is to be incorporated. Predictions of the separation location criteria incorporated in DONBOL are compared with the experimental data of Abeyounis (ref. 21) in figure 6(a). The large differences shown in predicted separation location can affect predicted afterbody pressure distributions and drag significantly, as illustrated in figure 6(b). Based on these limited results and because it more accurately predicts the location of separation on the steep, highly separated \( l/D = 0.8 \) boattail configuration, the method of Presz is recommended and is used for all further calculations presented in this paper.

An illustration of the capabilities of the present method for predicting the effects of free-stream Mach number on the pressure distribution and drag of an afterbody with separated boundary layer is shown in figure 7. The experimental data from references 22 and 23 shown in this figure are for the same \( l/D = 0.8, \ d_b/D = 0.51 \) circular-arc afterbody with solid cylindrical plume simulator for which separation location data are presented in figure 6(a). At a Mach number of 0.4 where the separation location is accurately predicted using the Presz criterion, the agreement between experimental and predicted pressure distributions is very good. As the difference in predicted and actual separation location increases with increasing free-stream Mach number, the agreement in pressure distributions between theory and experiment deviates somewhat. However, as shown in figure 7(b) the agreement between predicted and actual boattail pressure drag improves with increasing Mach number. This agreement is essentially within experimental accuracy throughout the range of Mach numbers for which the theory is applicable.

At a given free-stream Mach number, the agreement between theory and experiment is a function of boattail geometry. The comparisons between the theory and experiment of reference 23 shown in figure 8 indicate that for boattails with less closure than the configuration of figure 7, substantially better agreement between theory and experiment can result.
The capabilities of the present method for predicting the effects of Reynolds number on boattail pressure distributions and drag are illustrated in figure 9. The agreement between theory and experiment is a function of Reynolds number and boattail geometry. However, the predicted variation of the boattail-pressure drag coefficient with Reynolds number is in relatively good agreement with the experimental results (fig. 9(c)).

A comparison of the experimental (refs. 22 to 24) and predicted effects of the ratio of nozzle total pressure to free-stream static pressure NPR on the pressure distribution and drag of a i/D = 0.8, d^/D = 0.51 circular-arc nozzle is shown in figure 10. In general, the present method reasonably predicts the variation of the pressure distributions with NPR. However, at both M_B = 0.6 and 0.8 (figs. 10(a) and 10(b)) DONBOL generally predicts more positive pressures than actually exist on the nozzle. As a result, the magnitude of the boattail drag is substantially underpredicted (fig. 10(c)). These deficiencies probably result because the present method does not account for the effects of jet entrainment. Note that the present method does account for the effects of jet entrainment on the shape of the separation discriminating streamline, but does not account for jet plume entrainment in any manner downstream of the reattachment of the separated boundary layer. Jet entrainment downstream of reattachment should reduce the pressures on the nozzle and thereby increase the nozzle drag. As shown in figure 10(c), the present method accurately predicts the nozzle drag when the jet exhaust flow is simulated experimentally by a solid cylindrical sting. This solid sting, of course, does not simulate the effects of jet plume entrainment, but does simulate the effects of jet plume blockage on the flow over the nozzle. Even though the present method does not accurately predict the magnitude of nozzle drag, it does predict the decrease in drag at the higher nozzle total-pressure ratios. The present method does not, however, predict the increase in drag at the lower pressure ratios. Further illustrations of the capabilities of the present method for predicting pressure distributions and drag for nozzles with jet exhaust flow are shown in figure 11. Here the program DONBOL was used to calculate the flow over the equivalent bodies of reference 25 with the nozzles operating at an NPR of approximately 2.5. For these configurations, the predictions generally agree better with experiment than for the configuration shown in figure 10.

DESCRIPTION OF COMPUTER PROGRAM

A flow chart of program DONBOL is presented in figure 12 and a listing of the program is provided in the appendix. This program is written in overlay form and consists of the main overlay and four primary overlays. Primary overlays 1 to 3 are used to calculate the inviscid external flow, and overlay (5,0) is used to calculate the inviscid jet exhaust flow, the boundary-layer flow, and the "effective" body geometry for further iterations. The program uses nine disk files during computation. Input data are obtained from TAPE5 and the results are written on TAPE6 which is set equal to OUTPUT. A restart output file is written on TAPE7. The remaining disk files are used internally by the program. DONBOL requires about 125 000 octal storage locations on the Control Data CYBER 175 computer system and executes 15 iterations in approximately 3 minutes.
A brief description of the various routines in the program is given in the following list:

**DONBOL**  
This routine reads the input data, stores the x- and r-coordinates on TAPE13, and controls the iteration procedure. All primary overlays are called from this routine.

**ONE**  
This routine prints certain control parameter information and calls subroutines BASIC1 and MATRIX.

**BASIC1**  
This subroutine makes the compressibility correction transformations to the x- and r-coordinates, calculates coordinates of the midpoint of each body panel, and calculates the slope of each body panel.

**MATRIX**  
The influence coefficient matrix and the boundary condition matrix are set up in this subroutine. MATRIX calls subroutine XYZ.

**XYZ**  
The influence coefficients are calculated by this subroutine. A constant source of unit strength is assumed to act on each panel. The influence coefficient is the integral of the effect of the constant strength source. The subroutine calls XYZ1 and XYZ2.

**XYZ1**  
This subroutine performs the integration of the effects of the constant strength source for points within a specified radius of the singularity.

**XYZ2**  
This subroutine performs the integration using Simpson's rule to determine the influence of the unit source panel at all distances greater than the specified radius from the singularity. The routine calls subroutine BLIP.

**ELIP**  
This subroutine is used to calculate the value of various elliptical integrals.

**TWO**  
This routine initializes parameters for call to MISNA2.

**MISNA2**  
This subroutine calculates the strengths of the source panels by solving the matrix equation using a Seidel iteration procedure.

**THREE**  
This routine initializes various parameters and then calls subroutine AXIS. The pressure coefficients computed by AXIS are then written on TAPE13.

**AXIS**  
This subroutine calculates velocity components of the flow and surface-pressure coefficients. The velocity components are corrected for compressibility effects using either the Goethert or Labrujere method, before computing the pressure coefficient.

**FIVE**  
This routine is the interface between the inviscid external flow calculation and the viscous flow calculations. The body geometry and pressure coefficients are read from TAPE13. The inviscid jet plume
exhaust flow boundary and velocity are calculated. The viscous subroutine package is called to obtain boundary-layer parameters and the corrected effective body contour. See reference 8 for details of the subroutines in the viscous package. Routine FIVE also computes the drag coefficient. The results are printed and the final solution put on TAPE7 for further iteration if necessary.

IUNI This is a Langley Research Center computer system library subroutine. The subroutine uses first- or second-order Lagrangian interpolation to estimate the value of a set of functions at a specified value of the independent value.

Description of Input Data Cards

Sample input data required for program DONBOL are presented in figure 13. This figure presents the input data required to compute the flow over a boat-tail nozzle configuration with jet exhaust flow. This test case also illustrates the input data required to compute flow conditions at points off the body. Specifically, the input data required are as follows:

Card 1: identification.- Card 1 contains any desired identifying information in columns 1 to 80.

Card 2: control integers.- Card 2 contains 13 integers, each punched right justified in a five-column field. An identification of the card columns, the name used by the source program, and a description of each integer is given in the following table:

<table>
<thead>
<tr>
<th>Columns</th>
<th>FORTRAN name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1 to 5  | ISWITCH      | Calculation Option Code:  
If ISWITCH = 1, potential-flow solution only.  
If ISWITCH = 2, boundary-layer effects on pressure distribution are included in solution using an iteration scheme.  
If ISWITCH = 3, boundary-layer solution only. |
<p>| 6 to 10 | IPRINT       | Iteration number to start printing results. |
| 11 to 15| IPUNCH       | Punch option code: If IPUNCH greater than 0, last iteration is written on TAPE7 in format necessary for a restart of solution. CP for last iteration also written on TAPE7. |
| 16 to 20| ITERA        | Iteration number for first calculation of this submittal. For initial submittal of any calculation, ITERA must be 0. |</p>
<table>
<thead>
<tr>
<th>Columns</th>
<th>FORTRAN name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 to 25</td>
<td>ITERMAX</td>
<td>Maximum number of iterations (less than or equal to 20).</td>
</tr>
</tbody>
</table>
| 26 to 30 | IMACH        | Compressibility correction code:  
If IMACH = 1, Goethert compressibility correction used.  
If IMACH = 2, Labrujere compressibility correction used. |
| 31 to 35 | ISEP         | Separation location criteria code:  
If ISEP = 0, no separation model used.  
If ISEP = 1, separation location specified by user.  
If ISEP = 2, Presz control volume criterion used.  
If ISEP = 3, Goldschmied criterion used.  
If ISEP = 4, modified Page criterion used.  
If ISEP = 5, Stratford criterion used. |
| 41 to 45 | INT(3)       | X-array location to start search for separation. |
| 46 to 50 | INT(4)       | X-array location to end search for separation. |
| 51 to 55 | INT(5)       | Jet plume and entrainment option:  
If INT(5) = 0, omit jet plume and entrainment calculations.  
If INT(5) = 1, include jet plume and entrainment calculations. |
| 56 to 60 | INT(6)       | X-array location of nozzle exit. |
| 61 to 65 | INT(7)       | Smoothing parameter:  
If INT(7) = 0, no smoothing.  
If INT(7) = 1, aerodynamic body contour and pressure distribution are smoothed.  
INT(7) = 1 should be used. |
| 66 to 70 | IFLAG5       | An integer which if greater than 0 specifies that off-body points are to be calculated. |

Card 3: free-stream conditions and reference dimensions.—Card 3 contains quantities used to define the free-stream flow and dimensional information required to convert body coordinate inputs to meters. If the separation location is to be input by the user, it is given on this card. Identification of the card columns, names used in the source program, and a description of each variable is given in the following table:
Columns FORTRAN name Description
1 to 10 MO Free-stream Mach number.
11 to 20 PT Free-stream total pressure, Pa.
21 to 30 TT Free-stream total temperature, K.
31 to 40 REFL Reference length - factor required to convert input values of \(x\) and \(r\) to meters.
41 to 50 SREF Reference area, meters\(^2\).
51 to 60 XSEPND The \(x\)-coordinate of the separation location. Required if \(ISEP = 1\).

Card 4: jet exhaust conditions.- This card contains quantities used to define the jet exhaust flow. If there is no jet exhaust flow \((\text{INT}(5) = 0)\) this card may be blank, but it must be input. The card contains the following information:

Columns FORTRAN name Description
1 to 10 XMJET Mach number of jet at nozzle exit.
11 to 20 PTJET Jet total pressure, Pa.
21 to 30 TTJET Jet total temperature, K.
31 to 40 RJET Radius of nozzle exit.

Cards 5, 6, . . . : remaining data input cards.- The remaining data cards provide a description of the body geometry, the location of any off-body points at which the flow is to be calculated, and the surface pressure coefficients if the boundary-layer solution only is to be computed. Unless otherwise noted, each card contains up to six values with each value punched in a ten-column field with a decimal.

Body geometry cards: The first body geometry data card gives the number of coordinates, \(NN\). The integer, \(NN\) is punched in columns 1 to 5 right justified. The number of body coordinates may not be greater than 200. The next group of body geometry data cards contains the axial location at which the body radius is to be specified. There are exactly \(NN\) locations with up to six values per card. The next group of body geometry data cards contains the radius of the body at the specified axial locations. Again there are \(NN\) values of the body radius specified. Note that if the jet exhaust flow option is selected, an initial guess of the shape of the jet plume boundary must be included in the description of the body geometry.
Off-body points: If the flow is to be calculated at any off-body points and IFLAGS > 0, then the following cards must be input. First the number of off-body points must be specified on a data card. The number of off-body points is punched in columns 1 to 5 right justified. (Note that the sum of the points on the body of revolution and the off-body points may not be greater than 200.) Then a group of data cards giving the location of the x-coordinates at which the flow is to be calculated is input. This group of cards is followed by a group of cards on which the r-coordinates of the off-body points are specified.

Pressure coefficients cards: This group of cards is input only if the program is to be restarted or if ISWITCH = 3, that is, when the boundary-layer solution only is to be calculated. The pressure coefficient at each body x-coordinate location is input with six values per card.

Description of Output

Program output consists of printed output and a disk file TAPE7 written in the form necessary for a restart of the program. An example of the printed output is presented in figure 14 for the test case presented in figure 13.

The first page of output includes the program title, case identification, list of control options selected, free-stream conditions, and, if requested, jet exhaust flow conditions. On the second page, several diagnostic messages from various routines in the program are written.

Following these pages, the results of the calculation are output. Case identification and free-stream conditions are again specified. The iteration number, the reference length L, the reference area SREF, and the axial location of boundary-layer separation and reattachment are given. Following this information, tabulated listings of the body axial coordinate X/L, the body radial coordinate R/L, the body radius corrected for the discriminating streamline RDS/L, and the body radius corrected for boundary-layer displacement thickness and the discriminating streamline RC/L are printed. Also listed are values of pressure coefficient CP, local skin-friction coefficient CF, boundary-layer thickness DEL/L, boundary-layer displacement thickness DEL*X/L, boundary-layer momentum thickness THETA/L, and boundary-layer shape factor H. In addition, listings of the pressure drag coefficient CDP, skin-friction drag coefficient CDF, and total drag coefficient CDT are given. The drag values listed are based on the reference area SREF and are the integrals of the pressure forces and/or skin-friction forces from the nose of the body to the specified X/L location. To obtain the nozzle boattail pressure drag coefficient, for example, it is necessary to subtract the value of the pressure drag coefficient at the start of the boattail from the value of the pressure drag coefficient at the nozzle exit or end of the boattail. This information is repeated for each iteration as specified in the input data.

If flow conditions at off-body points are calculated, the axial location X/L and radial location R/L of the off-body points are tabulated on the next page together with the ratio of axial velocity to free-stream velocity VX.
the ratio of radial velocity to free-stream velocity $V_R$, and the ratio of local velocity to free-stream velocity $V_T$. Also tabulated are the local flow angle $\eta$, the local Mach number $M_L$, and the local pressure coefficient $C_P$.

CONCLUDING REMARKS

A computer program has been written to compute the flow over axisymmetric nozzle configurations at subsonic speeds with and without separated flow. The computer algorithm is based on a patched viscous-inviscid interaction procedure. That is, solutions for the various regions of the flow are coupled together and solved iteratively to obtain a converged solution. The results of the present algorithm called DONBOL are in good agreement with experimental pressure distribution results for flow over nozzles with the jet exhaust simulated with solid bodies. The method substantially underpredicts the magnitude of the boattail drag when the jet exhaust flow is simulated with high-pressure air. This deficiency results because the present technique does not account for the effects of jet plume entrainment downstream of reattachment of the separated boundary layer on the flow over the nozzle. The method is limited to free-stream Mach numbers below that for which flow on the body of revolution reaches sonic speeds.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
April 11, 1979
APPENDIX

TABULATED LISTING OF COMPUTER PROGRAM
APPENDIX

OVERLAY(LINK,0,0)

PROGRAM DONBOL( INPUT = 201, OUTPUT = 201, TAPE5, TAPE6, OUTPUT, TAPE3 = 1001, DON
1 TAPE4 = 1001, TAPE9 = 1001, TAPE11 = 1001, TAPE12 = 1001, TAPE13 = 1001, TAPE7 = 1000
201)

*****************************************************

INPUT DATA DESCRIPTION FOR PROGRAM DONBOL

*****************************************************

<table>
<thead>
<tr>
<th>CARD</th>
<th>COL</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-80</td>
<td>HEADER</td>
<td>CASE DESCRIPTION</td>
</tr>
<tr>
<td>2</td>
<td>1-5</td>
<td>ISWITCH</td>
<td>CALCULATION OPTION CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISWITCH = 1 POTENTIAL FLOW SOLUTION ONLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISWITCH = 2 BOUNDARY LAYER EFFECTS ON SOLUTION USING AN ITERATION SCHEME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISWITCH = 3 BOUNDARY LAYER SOLUTION ONLY</td>
</tr>
<tr>
<td>2</td>
<td>6-10</td>
<td>IPRINT</td>
<td>ITERATION NUMBER TO START PRINTING RESULTS</td>
</tr>
<tr>
<td>2</td>
<td>11-15</td>
<td>IPUNCH</td>
<td>PUNCH OPTION CODE = IF IPUNCH GREATER THAN 0 LAST ITERATION ON TAPE 7 IN FORMAT NECESSARY FOR A RESTART OF SOLUTION CP FOR LAST ITERATION ALSO WRITTEN ON TAPE 7</td>
</tr>
<tr>
<td>2</td>
<td>16-20</td>
<td>ITERA</td>
<td>ITERATION NUMBER FOR FIRST CALCULATION OF THIS SUBMITTAL</td>
</tr>
<tr>
<td>2</td>
<td>21-25</td>
<td>ITERMAX</td>
<td>MAXIMUM NUMBER OF ITERATIONS (LESS THAN OR EQUAL TO 20)</td>
</tr>
<tr>
<td>2</td>
<td>26-30</td>
<td>IMACH</td>
<td>COMPRESSIBILITY CORRECTION CODE =</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF IMACH = 1 GOETERS COMPRESSIBILITY CORRECTION USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF IMACH = 2 LABRUIERE COMPRESSIBILITY CORRECTION USED</td>
</tr>
<tr>
<td>2</td>
<td>31-35</td>
<td>ISEP</td>
<td>SEPARATION LOCATION CRITERIA CODE =</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 0 NO SEPARATION MODEL USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 1 SEPARATION LOCATION SPECIFIED BY USER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 2 PREZ CONTROL VOLUME CRITERIA USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 3 GOLDBEMMELER CRITERIA USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 4 MODIFIED PAGE CRITERIA USED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IF ISEP = 5 STRATFORD CRITERIA USED</td>
</tr>
<tr>
<td>2</td>
<td>36-40</td>
<td></td>
<td>NOT USED</td>
</tr>
<tr>
<td>2</td>
<td>41-45</td>
<td>INT(3)</td>
<td>X-ARRAY LOCATION TO START SEARCH FOR SEPARATION</td>
</tr>
</tbody>
</table>
| 2    | 46-50| INT(4) | X-ARRAY LOCATION TO END SEARCH FOR
APPENDIX

C

2 51=55 INT(5) JET PLUME AND ENTRAPMENT OPTION =
     IF INT(5)=0 OMIT JET PLUME AND
     ENTRAPMENT CALCULATIONS,
     IF INT(5)=1 INCLUDE JET PLUME AND
     ENTRAPMENT CALCULATIONS,

2 56=60 INT(6) X-ARRAY LOCATION OF NOZZLE EXIT.

2 61=65 INT(7) SMOOTHING PARAMETER =
     IF INT(7)=0 NO SMOOTHING,
     IF INT(7)=1 AERODYNAMIC BODY CONTOUR AND
     PRESSURE DISTRIBUTION ARE SMOOTHED.

2 66=70 IFLAGS AN INTEGER WHICH IF GREATER THAN 0
     SPECIFIES THAT OFF BODY POINTS ARE TO
     BE CALCULATED.

CARD COL NAME DESCRIPTION

3 1=10 MN FREE STREAM MACH NUMBER.

3 11=20 PT FREE STREAM TOTAL PRESSURE, PASCALS

3 21=30 TT FREE STREAM TOTAL TEMPERATURE, KELVIN

3 31=40 REPL REFERENCE LENGTH = FACTOR REQUIRED TO
     CONVERT INPUT VALUES OF X AND R TO
     METERS.

3 41=50 SRF Reference Area, 50 Meters

3 51=60 SXPND THE X-COORDINATE OF THE SEPARATION
     LOCATION. REQUIRED IF ISEP=1.

4 1=10 XMJET MACH NUMBER OF JET AT NOZZLE EXIT.

4 11=20 PTJET JET TOTAL PRESSURE, PASCALS

4 21=30 TTJET JET TOTAL TEMPERATURE, KELVIN

4 31=40 RJET RADIUS OF NOZZLE EXIT

5 1=5 NN NUMBER OF COORDINATES FOR BODY

6 1=60 X(I),I=1,NN THE X-COORDINATES OF THE POINTS DEFINING
     THE BODY. DATA IS INPUT WITH A
     FORMAT OF 6F10.6, MAY BE MORE THAN
     ONE CARD

7 1=60 R(I),I=1,NN THE R-COORDINATES OF THE POINTS DEFINING
     THE BODY. DATA IS INPUT WITH A
     FORMAT OF 6F10.6, MAY BE MORE THAN
     ONE CARD

IF IFLAGS GREATER THAN 0 THE FOLLOWING CARDS MUST BE INPUT

CARD COL NAME

8 1=5 NN NUMBER OF OFF BODY POINTS

9 1=60 X(I),I=1,NN THE X-COORDINATES OF THE OFF BODY

DON 66
DON 67
DON 68
DON 69
DON 70
DON 71
DON 72
DON 73
DON 74
DON 75
DON 76
DON 77
DON 78
DON 79
DON 80
DON 81
DON 82
DON 83
DON 84
DON 85
DON 86
DON 87
DON 88
DON 89
DON 90
DON 91
DON 92
DON 93
DON 94
DON 95
DON 96
DON 97
DON 98
DON 99
DON 100
DON 101
DON 102
DON 103
DON 104
DON 105
DON 106
DON 107
DON 108
DON 109
DON 110
DON 111
DON 112
DON 113
DON 114
DON 115
DON 116
DON 117
DON 118
DON 119
DON 120
DON 121
DON 122
DON 123
DON 124
DON 125
DON 126
DON 127
DON 128
DON 129
DON 130
APPENDIX

**APPENDIX**

POINT8, DATA IS INPUT WITH A FORMAT OF 6F10.6. MAY BE MORE THAN ONE CARD.

10 I=60 R(I),I=1,NN THE R-COORDINATES OF THE OFF BODY POINT8, DATA IS INPUT WITH A FORMAT OF 6F10.6. MAY BE MORE THAN ONE CARD.

IF ISWITCH IS EQUAL TO 3 THE FOLLOWING CARD MUST BE INPUT

CARD COL NAME DESCRIPTION
11 I=60 CP(I),I=1,NN PRESSURE COEFFICIENT AT EACH X-COORDINATE ON BODY. DATA IS INPUT WITH A FORMAT OF 6F10.6. MAY BE MORE THAN ONE CARD.

**APPENDIX**

C

COMMON HEDR(8),ISWITCH,IPUNCH,ITERA,ITERMAX,IMACH,ISEP,INT(8),IFLAG,105,FLG05,MN,APT,ATT,REFL,BREF,XSEPND,NN,BDN,ND(2),NT,N8IGA,IPRINT,DOM
2AMJET,PTJET,TTJET,RJET,RSTAR
COMMON /SAVE/ VDOM(951)
DIMENSION X(200),R(200),CP(200)
INTEGER FLG05,BDN
REAL MN

LINK=ALLINK
BDN=1
DO 10 I=1,200
10 CP(I)=.0

C

READ (5,70) HEDR
IF (EOF(5)) 60,30

C

READ (5,80) ISWITCH,IPRINT,IPUNCH,ITERA,ITERMAX,IMACH,ISEP,INT(I)
READ (5,90) MN,APT,ATT,REFL,BREF,XSEPND,AMJET,PTJET,TTJET,RJET
READ (5,100) NN
READ (5,110) (X(I),I=1,NN)

C

IF (I.EQ.1.OR.ITERA.GT.0) READ (5,120) (R(I),I=1,NN)
WRITE (13) NN,(X(I),I=1,200)

C

CONTINUE

IF ((ISWITCH,EQ,3).OR.ITERA.GT.0) READ (5,90) (CP(I),I=1,NN)
WRITE (13) (CP(I),I=1,200)
IF (ISWITCH,EQ,3) GO TO 50
IF (ISWITCH,EQ,1) OR (ISWITCH,EQ,2) GO TO 50
GO TO 20

C

REWIND 3
REWIND 4
REWIND 9
REWIND 11
REWIND 12
REWIND 13

C

CONTINUE
STOP
C

DON 131
DON 132
DON 133
DON 134
DON 135
DON 136
DON 137
DON 138
DON 139
DON 140
DON 141
DON 142
DON 143
DON 144
DON 145
DON 146
DON 147
DON 148
DON 149
DON 150
DON 151
DON 152
DON 153
DON 154
DON 155
DON 156
DON 157
DON 158
DON 159
DON 160
DON 161
DON 162
DON 163
DON 164
DON 165
DON 166
DON 167
DON 168
DON 169
DON 170
DON 171
DON 172
DON 173
DON 174
DON 175
DON 176
DON 177
DON 178
DON 179
DON 180
DON 181
DON 182
DON 183
DON 184
DON 185
DON 186
DON 187
DON 188
DON 189
DON 190
DON 191
DON 192
DON 193
DON 194
DON 195
DON 196
APPENDIX

70 FORMAT (8A10) DON 197
80 FORMAT (1615) DON 199
90 FORMAT (6F10.6) DON 199
END DON 200*
OVERLAY(LINK,1,0)
PROGRAM ONE
C
CONTROL FOR BASIC DATA AND FORM MATRIX ONE
COMMON HEDR(8),ISWITCH,IPUNCH,ITERA,ITERMAX,IMACH,ISEP,INT(8),IFRAME
185,FLG05,MN,APT,ATT,REFL,SRFL,XT3EPND,NN,BDN,ND(2),NT,NSIGA,IPRINT,ONE
2AMJET,PTJET,TTJET,RJET,RTAR
COMMON /CL/ XI(200),YI(200),X2(200),Y2(200),DEL3(200),XINA(200),CONF
1SA(200),XP(200),YP(200)
COMMON /TL/ TX1(200),TY1(200),NG(200),TG(200),ALFA(200),R3DS(200),ONE
1DALT(200),TEMP(1017)
INTEGER FLG05,BDN
REAL MN,NG
C
OUTPUT CASE CONTROL DATA
IF (ITERA.GT.0) GO TO 10
WRITE (6,250) ONE 18
WRITE (6,20) HEDR ONE 19
IF (IFLAG5.GT.0) WRITE (6,50) ONE 20
IF (IMACH.EQ.1) WRITE (6,00) ONE 21
IF (IMACH.EQ.2) WRITE (6,50) ONE 22
WRITE (6,60) ONE 23
IF (ISEP.EQ.0) WRITE (6,70) ONE 24
IF (ISEP.GT.0) WRITE (6,80) ONE 25
WRITE (6,90) X8EPND ONE 26
WRITE (6,100) ONE 27
WRITE (6,110) ONE 28
WRITE (6,120) ONE 29
WRITE (6,130) ONE 30
WRITE (6,140) INT(3) ONE 31
WRITE (6,150) INT(4) ONE 32
WRITE (6,160) ONE 33
WRITE (6,170) INT(6) ONE 34
WRITE (6,180) ONE 35
WRITE (6,190) ONE 36
WRITE (6,200) ONE 37
WRITE (6,210) MN,APT,ATT ONE 38
WRITE (6,220) RN ONE 39
IF (INT(5).GT.0) WRITE (6,240) ONE 40
IF (INT(5).GT.0) WRITE (6,210) AMJET,PTJET,TTJET
WRITE (6,250) ONE 41
CALL BASIC
N8IGA=1 ONE 42
REWIND 4 ONE 43

C
SETUP FOR UNIFORM FLOW
10 CALL BASIC
N8IGA=1 ONE 44
REWIND 4 ONE 45

APPENDIX

CALL MATRIX ONE 62
C ONE 63
C ONE 64
C ONE 65
20 FORMAT (10X,6HDONBOL == AN AXI-SYMMETRIC INVISCID/VISCID INTERACTION PROGRAM/16X,52HBY LAWRENCE E. PUTNAM, NASA, LANGLEY RESEARCH CENTER/2X,13HCASE TITLE = ,BA10/13X,29H**** CASE CONTROL DATAGONE 68
3 ****/Y) ONE 69
30 FORMAT (13X,15H5FF•BODY POINTS) ONE 70
40 FORMAT (13X,35HGOETHERT COMPRESSION CORRECTION) ONE 71
50 FORMAT (13X,36HLAURELLE, COMPRESSION CORRECTION) ONE 72
60 FORMAT (13X,46HMOODIFIED REMOTKD TUCKER BOUNDARY LAYER SOLUTION) ONE 73
70 FORMAT (13X,29HSEP. MORE SEPARATED FLOW MODEL NOT USED) ONE 74
80 FORMAT (13X,64HPRESZ MODIFIED CONTROL VOLUME DISCRIMINATING STREAMLINE SOLUTION) ONE 75
90 FORMAT (13X,46HPRESZ MODIFIED CONTROL VOLUME SEPARATION LOCATION SPECIFIED BY USER AT X/L =F10,ONE 77
91 FORMAT (13X,46HPRESZ CONTROL VOLUME SEPARATION LOCATION CRITERIA) ONE 78
100 FORMAT (13X,46H4PREBZ CONTROL VOLUME SEPARATION LOCATION CRITERIA) ONE 79
110 FORMAT (13X,40HGOETHERT SEPARATION LOCATION CRITERIA) ONE 80
120 FORMAT (13X,42HMOODIFIED PAGE SEPARATION LOCATION CRITERIA) ONE 81
130 FORMAT (13X,48HSTRAFORD SEPARATION LOCATION CRITERIA) ONE 82
140 FORMAT (13X,34HSTART SEARCH FOR SEPARATION AT I =I4) ONE 83
150 FORMAT (13X,32HEND SEARCH FOR SEPARATION AT I =I4) ONE 84
160 FORMAT (13X,29HJET EXHAUST PLUME CALCULATION) ONE 85
170 FORMAT (13X,18HNOZZLE EXIT AT I =I4) ONE 86
180 FORMAT (13X,26HSMOOTH AERODYNAMIC CONTOUR) ONE 87
190 FORMAT (13X,28HSMOOTH PRESSURE DISTRIBUTION) ONE 88
200 FORMAT (140,12X,22HPRESSURE STREAM CONDITIONS) ONE 89
210 FORMAT (20X,20HMACII NUMBER =F12.3/20X,20HTOTAL PRESSURE ONE 90
211 FORMAT (20X,20HMACII NUMBER =F12.3/20X,20HTOTAL TEMPERATURE =F12.3/7H KELVIN) ONE 91
220 FORMAT (20X,20HMACII NUMBER =F12.3,18H MILLION PER METER) ONE 92
230 FORMAT (20X,20HMACII NUMBER =F12.3) ONE 93
240 FORMAT (140,12X,22HJET EXHAUST CONDITIONS AT NOZZLE EXIT) ONE 94
250 FORMAT (13X,29HJET EXHAUST CONDITIONS AT NOZZLE EXIT) ONE 95
END ONE 96
SUBROUTINE BASIC ONE 97
C ONE 98
COMMON HEOR(8),(I8WITCH,IPUNCH,ITERA,ITERMAX,IMACH,ISEP,INT(8),IFLAGS,BN ONE 99
COMMON XCLX X1(200),Y1(200),X2(200),Y2(200),DEL8(200),81NA(200),COBA ONE 100
COMMON /TL/ TX1(200),TY1(200),NG(200),T6(200),ALFA(200),R8D8(200) BAS ONE 101
INTEGER FLG05,BDN BAS 102
REAL MN,NG BAS 103
C ONE 104
REMIND 13 BAS 105
NT 0 BAS 106
K=0 BAS 107
K2=1 BAS 108
IF (ITERA,GT,ITERMAX) FLG05=IFLAGS BAS 109
IF (FLG05,NE,0) K2=2 BAS 110
C ONE 111
* MAJOR LOOP • NO. OF BODIES • OFF BODY POINTS BAS 112
C ONE 113
DO 130 L=1,K2 BAS 114
IF (FLG05,GT,0,AND,L,GT,1) GO TO 10 BAS 115
NO(L),NNN BAS 116
M=NNN=1 BAS 117
READ (13) BLANK BAS 118
READ (13) BLANK BAS 119
READ (13) INT1(I),IM1,NNN BAS 120
READ (13) INT1(I),IM1,NNN BAS 121
22
APPENDIX

GO TO 20
10 CONTINUE
REWIND 12

C * SAVE SINA AND COSA ON TAPE & FOR CALC. OF MATRIX SOLUTION (RIGHT HAND MATRIX)
C
C WRITE (4) (SINA(I),I=1,NT),(COSA(I),I=1,NT)
C RETURN
C
150 FORMAT (215)
160 FORMAT (6F10.0)
END
SUBROUTINE MATRIX

* COMPUTE MATRIX A,B,Z OR X,Y,Z

COMMON HEDR(6),ISWITCH,IPUNCH,ITERA,ITERMAX,IMACH,ISEP,INT(8),IFLAG,MAT
1G5,FLGOS,MTN,AP,T,ATT,REFL,BREF,XBEPND,NM,BDN,ND(2),NT,NGA,IPRINT,MAT
2AMJET,PTJET,TTJET,RJET,RSTAR
COMMON /CL/ XI(200),VI(200),X2(200),Y2(200),DELS(200),SINA(200),COMAT
1SA(200),XP(200),YP(200)
COMMON /TL/ A(200),B(200),AX(200),AY(200),AZ(200),CX(200),CY(200),MAT
1CT(200),AXV(200),AYV(200),VN(200,1),VT(200,1),BDN,YZERO,IAIC,II,JJ,MAT
2J,LJ,DS,DX,DY,NX,NY,NZ,EX,EEK,KK
INTEGER FLGOS,BDN
REAL MN

* INITIALIZE

L1=NT
MON0=0
YZERO=0.0
IAC=1
10 DO 20 I=1,NT
J=1
VN(I,J)=0.0,
VT(I,J)=0.0,
20 DO 70 I=1,L1
II=I

* I MIDPOINT LOOP

DO 70 I=1,ND
J1=THE COORDINATE COUNTER

J1=0
N1=ND(I)+1
KK=1
DO 30 J=1,N1
JJ=J
J1=J1+1

* COMPUTE X,Y,Z MATRICES

CALL XYZ
CONTINUE
30 J1=J1+1
IF (BON) 40,50,40

* SAVE X,Y,Z ON TAPE *OFF BODY POINTS
40 WRITE (9) (AX(J),J=1,NT),(AY(J),J=1,NT),(AZ(J),J=1,NT)
GO TO 70

* SAVE A,B,Z ON TAPE *ON BODY
50 DO 60 J=1,NT
A(J)=AX(J)*SINA(I)+AY(J)*COSA(I)
60 CONTINUE
APPENDIX

60  B(J)*MAX(J)*COSA(I)+AY(J)*SINE(I)
write (9) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)
70  continue
70  test if off body completed
70  test if off body
70  if (FLG05.EQ.0,.OR.,BQN.NE.0.) go to 90
70  * initial for off body * then re-enter i,j loops

80  BON=1,
80  L1=ND(2)
80  go 80 if(i=l,l1
80  y2(i)=xp(i)
80  go to 10
80  rewind 9
80  rewind 4
80  return
80  end

90  subroutine XYZ

40  * control for x,y,z matrices computation
40  common /cl/x1(200),y1(200),x2(200),y2(200),del8(200),sine(200),cox(200)
40  common /tl/a(200),r(200),ax(200),ay(200),az(200),cx(200),cy(200)
40  integer flg05, bdn
40  real mn

50  if (bdn) 50,10,50
50  if (j.i) 60,20,60

50  j equal i path
50  call XYZ1
50  go to 190
50  * j not equal i path
50  compute minimum distance to i midpoint
50  go to 180

60  * initial y coordinate mid-point for zero test
60  yzero=y2(i)*0.000001
60  * compute minimum distance to i midpoint
60  d1=(y2(i)=y1(j1))**2+(y2(i)=y1(j1))^2
APPENDIX

N2*(X2(I)+X2(J))**2+(Y2(I)+Y2(J))**2
N3*(X3(I)+X3(J+1))**2+(Y3(I)+Y3(J+1))**2
IF (D1=D2) 80,80,70
70 IF (D2=D3) 100,100,90
80 DM=8QRT(D3)
GO TO 120
90 DM=8QRT(D2)
GO TO 120
100 DM=8QRT(D1)

* COMPUTE NO. OF INTERVALS(NI) AND DELTA S (DS)
FOR SIMPSON RULE INTEGRATION

120 TF (DM.EQ.0,0) GO TO 150
NI=6.*DEL8(J)/DM*0.9
TF (NI) 130,130,140
130 NI=3
DS=DEL8(J)/2,
GO TO 170
140 NI=NI+1
TF (NI=128) 160,150,150
150 NI=129
NI=NI+1
DS=DEL8(J)/128,
GO TO 170
160 XNI=NI
NI=NI+1

* COMPUTE X,Y,Z MATRICES FOR SJ LESS THAN OR EQUAL .08
COMMON HEDR(8),ISWITCH,IPUNCH,ITERA,ITERMAX,IMACH,IBEP,INT(8),IFLAXY
105,FLG5,8,APT,ATT,REFL,BREF,X8EPND,NN, BDN,ND(2),NT,8BIGA,IPRINT,XY
2AMJET,PTJET,TTJET,RJET,R8TAR
COMMON /CL/ X1(200),Y1(200),X2(200),Y2(200),DEL8(200),SINA(200),CDXY
18A(200),XP(200),YP(200)
COMMON /TL/ A(200),B(200),AX(200),AY(200),AZ(200),CX(200),CY(200),CZ(200)
1CZ(200),AXV(200),AYV(200),AVN(200),AYV(200),AXV(200),AYV(200),AXV(200)
2,J,D8,OX,DY,NJ,XJ,YJ,XX,EEK,K
INTEGER FLGS,BDN
REAL MN

* INITIALIZE

T1=SJ=8J
T2=ALOG(SJ/8.0)
T3=BINA(J)+SINA(J)
T4=T2+T3
T5=6666666667*T3
T6=T5*T3
T7=B5+8J
T8=T7+T7
T9=B2831853*COSA(J)
T10=B2831853*SINA(J)
T11=T1+8J

* AXIS FLOW
APPENDIX

10  AX(J) = T10 * SINA(J) * COSA(J) * (T7 + (T4 + 2.1666667) * T11/12)  XY1 32
    AX(J) = T10 * SINA(J) * COSA(J) * (T7 + (T4 + 2.1666667) * T11/12)  XY1 33
    AY(J) = T7 + T4 * T9 * (1 + T2 + T3) * T6 * T11/6,  XY1 34
    T12 = T7 + T1  XY1 35
    AZ(J) = Y2(J) * T8 * (1 + T2 + T1 * (2 + T12 + 3) + T2 * (1 + T12)) / 144,  XY1 36
    RETURN  XY1 37
    END  XY1 38
SUBROUTINE XYZ2  XY2 1
C

* COMPUTE X, Y, Z MATRICES USING SIMPSON RULE INTEGRATION  XY2 3
C
COMMON HEDR(8), ISWITCH, IPUNCH, ITERA, ITERMAX, IMACH, ISEP, INT(8), IFLA  XY2 5
15, FLO05, MN, APT, ATT, REF, SREP, XSEPND, NN, RDN, ND(2), NT, NBIGA, IPRINT, XY2 6
2XJET, PTJET, TTJET, RSTAR  XY2 7
COMMON /CL/ X1(200), Y1(200), Z1(200), Y2(200), Y2(200), DEL(200), SINA(200), COSA(200), XY2 8
1A(200), XP(200), YP(200)  XY2 9
COMMON /TL/ A(200), B(200), AX(200), AY(200), AZ(200), CX(200), CY(200), XY2 10
1T(200), AXV(200), AYV(200), AV(200), V(200), V(200), 1, BON, YZERO, IAC, I, J, IXY2 11
2, 8J, 08, X, DY, NI, XJ, YJ, XK, EK, EKK, K
INTEGER FLOG5, RDN
REAL MN

* INITIALIZE  XY2 16

10  B2 = 6.66666667 * DS  XY2 18
    DS = DS + 82  XY2 19
    T1 = Y2(I) * YP(I)  XY2 20

* NO. OF INTERVAL LOOP  XY2 21

01  100  I8 = 1, NI  XY2 23
    XJ = XJ + DX  XY2 24
    YJ = YJ + DY  XY2 25
    T2 = YJ / YJ  XY2 26
    T3 = X2(I) = XJ  XY2 27
    T4 = T3 * T3  XY2 28
    T5 = Y2(I) * YJ * YJ  XY2 29
    T6 = T4 * T5  XY2 30
    T7 = SQRT(T6)  XY2 31
    T8 = T2 * T4  XY2 32
    T9 = Y2(I) * YJ * YJ  XY2 33
    T10 = T9 * T4  XY2 34

* COMPUTE ELLIPTIC INTEGRAL  XY2 35

XX = YJ = Y2(I) / T6
CALL ELIP

* AXIS FLOW  XY2 41

T1 = YJ / T7
1F (Y2(I), EQ. 0.) GO TO 20  XY2 42
T12 = YJ / Y2(I)  XY2 43
FV2 = (EK + EK) * (T1 = T8) / T7  XY2 44
FV3 = Y2(I) / T10 * T3 / T7 * EK  XY2 45
F1 = FV3 * T12  XY2 46
F2 = FV2 * T12  XY2 47
F4 = FV2 * T3 / Y2(I)  XY2 48
G0 TO 30  XY2 49

20  FV2 = 0.  XY2 50
    FV5 = 0.  XY2 51
    FV8 = 0.  XY2 52
    F2 = 0.  XY2 53
    F1 = T11 / T8 * T3 * EK  XY2 54
    T3 = T11 * EKK  XY2 55

C
APPENDIX

C * SIMPSON RULE INTEGRATION
C
C IF (18.1) 40, 40, 50
C
40 AXS+F1
AYS+F2
AZS+F3
TA=0
GO TO 120
50 IF (18=N1) 60, 90, 60
60 IF (1A) 80, 70, 80
C
C * FIRST PASS
C
70 AXS+AXS+4, *F1
AYS+AY8+4, *F2
AZS+AZ8+4, *F3
TA=1
GO TO 120
C
C * EVEN PASS
C
80 AXS+AXS+F1+F1
AYS+AY8+F2+F2
AZS+AZ8+F3+F3
TA=0
GO TO 120
C
C * ODD PASS
C
90 IF (J=I) 110, 100, 110
100 IF (BON, NE, 0, 0) GO TO 110
AX(J)=AX(J)+8*(AXS+F1)
AY(J)=AY(J)+8*(AY8+F2)
AZ(J)=AZ(J)+8*(AZ8+F3)
GO TO 120
110 AX(J)=8*(AXS+F1)
AY(J)=8*(AY8+F2)
AZ(J)=8*(AZ8+F3)
120 CONTINUE
130 CONTINUE
RETURN
END
SUBROUTINE ELIP

C * HASTINGS APPROXIMATION FOR ELLIPTIC INTEGRALS
C
C COMMON / TL, A(200), B(200), AX(200), AY(200), AZ(200), CY(200), CY(200), ELI
1 CX(200), AXV(200), AVV(200), VM(200, 1), WT(200, 1), BON, YZERO, IAC, I, J, IELI
2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
C
10 ETA=1.*XK
IF (ETA) 20, 20, 30
20 WRITE (6, 40) ETA
ALL EXIT
30 ELN=ALOG(ETA)
EKK=1, 38629436112*ETA*(0, 0966634259*ETA*(0, 0339092383*ETA*(0, 0376758776*ETA*(0, 0328355346*ETA*(0, 0441780712)))))
ELN=0.5*ETA*(0.12498593597*ETA*(0.03328355346*ETA*(0.0441780712)))
EKK=0.443251463*ETA*(0.06260601220*ETA*(0.04757383546*ETA*1.0136506451))=ELN*(ETA*0.2498366310+ETA*(0.04200180057+ETA*ELI)
2(0.0406975726*ETA*0.0026444639)))
RETURN
ELI 1
ELI 2
ELI 3
ELI 4
ELI 5
ELI 6
ELI 7
ELI 8
ELI 9
ELI 10
ELI 11
ELI 12
ELI 13
ELI 14
ELI 15
ELI 16
ELI 17
ELI 18
ELI 19
ELI 20
ELI 21
APPENDIX

40 FORMAT (1X,13H,27H* ERROR IN SUBROUTINE ELIP ,ETA=F15.8) ELI 22
END
OVERLAY(LINK,2,0)
PROGRAM TWO

C
* COMPUTE SOURCE DENSITY SIGMA BY SIEDEL ITERATION
C
COMMON HEDR(8), ISWITCH, IPUNCH, ITERA, ITERMAX, IMACH, ISEP, INT(8), IFLATWO 10
16S, FLGOS, MN, APT, ATT, REFI, SREF, XSEPND, NN, BDN, ND(2), NT, NSIGA, IPRINT, TWO 20
COMMON /C2/ A(200), R(200), NSIG, IT
DIMENSION SIGA(200,1)
INTEGER FLGOS, BDN
REAL MN

C
* AXIS FLOW
C
READ (4) (R(I), I=1, NT)
REWIND 4
ITER 0
NSIG=NSIGA

C
* SOLVE SIMULTANEOUS EQUATIONS FOR SIGMAs
CALL MISNA2 (SIGA)
REWIND 9

C
* WRITE SIGMAS ON TAPE 3
WRITE (3) (SIGA(I), I=1, NT)
END
SUBROUTINE MISNA2 (SIGA)

C
* SOLVE LINEAR SIMULTANEOUS EQUATIONS BY SIEDEL ITERATION
COMMON HEDR(8), ISWITCH, IPUNCH, ITERA, ITERMAX, IMACH, ISEP, INT(8), IFLATWO 10
16S, FLGOS, MN, APT, ATT, REFI, SREF, XSEPND, NN, BDN, ND(2), NT, NSIGA, IPRINT, MIS 20
COMMON /C2/ A(200), R(200), NSIG, IT
DIMENSION SIGA(200,1), KFLAG(1), DSIG(200,1) MIS 30
INTEGER FLGOS, BDN
REAL MN

C
* INITIALIZE
10 NTU=0
ITER=0
NCONV=0
DO 20 J=1, NSIG
KFLAG(J)=0
DO 20 I=1, NT
SIGA(I,J)=0.0
20 20
DO 30 I=1, NSIG
DO 40 J=1, NT
DSIGA(I,J)=0.0
30 30
DO 40
40 40

C
* COMPUTE SIGMA AND DELTA SIGMA
DO 100 I=1, NT
IF (NTU=3) 50, 60, 70
C
* PLACE A IN LEFT SIDE MATRIX
50 READ (9) (A(L), L=1, NT)
C
* SAVE LEFT SIDE MATRIX
APPENDIX

WRITE (3) (A(L),L=1,NT)
WRITE (11) (A(L),L=1,NT)
GO TO 80

* READ LEFT SIDE MATRIX
READ (3) (A(L),L=1,NT)
GO TO 80
READ (11) (A(L),L=1,NT)
GO TO 80

DO 100 J=1,NBIG
IF (KFLAG(J),NE,0) GO TO 100
SUM=0.0
DO 90 L=1,NT
SUM=SUM+A(L)*SIG(L,J)
816(I,J)=(R(I)-SUM)/A(I)
SIG(I,J)=SIG(I,J)+D8IG(I,J)
90 CONTINUE

CONTINUE

REWIND 3
REWIND 11
ITER=ITER+1
DO 110 J=1,NBIG
IF (KFLAG(J),NE,0) GO TO 110
IF (D8IG(J,GE,1.E)6) GO TO 110
KFLAC(J)=ITER
NCONV=NCONV+1
IF (NCONV,LE,1) GO TO 130
CONTINUE

IF (ITER.EQ.100) GO TO 130
IF (NTU.LE.10) GO TO 120
NTU=NTU+1
GO TO 10

CONTINUE

WRITE (6,160) ITERA
GO TO 150
WRITE (6,170) ITERA, KFLAG(J)
CONTINUE
RETURN
FORMAT (1HQ,10HFOR lTERA«, I 3, 46H NO
10 ITERATIONS)
FORMAT (1HO,10HFOR lTERA«, IS. IS, 46H
1GENCE IN MI8NA2)
END

* COMPUTE VELOCITY COMPONENTS AND PRINT

COMMON HEDR(8),ISWITCH,IPUNCH,ITERA,ITERMAX,IMACH,IBEP,INT(8),IFL,THR 5
105,FLGO5,WN,APT,ATT,REF,SBREF,SBPNO,NBN,NOC2,NT,NSIG,IPRINT,THR 6
2AMJET,PTJET,TJET,JET,STAR
COMMON /C1/ X1(200),Y1(200),X2(200),Y2(200),DEL8(200),SUM(200),COTHM 8
1HA(200),XP(200),YP(200)
COMMON /TC/ RB(200,2),SIG(200,1),A(200),B(200),Z(200),PMI(200,1),XTHR 10
IN(200,1),T(200,1),TS(200,1),NSIG,NP,NI,SUMV,SUM(4)
APPENDIX

```
INTEGER FLG05, BDN
REAL MN

C
REWRITE

IF (FLG05, EQ, 0) GO TO 10

* READ OFF-BODY XP, YP

NP=END(2)
READ (12) (XP(I), I=1, NP), (YP(I), I=1, NP)

* READ X1, Y1, X2, Y2, DEL8 WITH MACH NO. ADJUSTMENT IF ANY

10 NI=NT+1
READ (12) (XI(I), I=1, NI), (YI(I), I=1, NI), (X2(I), I=1, NT), (Y2(I), I=1, NT), (DEL8(I), I=1, NT)

* READ SINA, COSA, NO. TO...

READ (4) (A(I), I=1, NT), (B(I), I=1, NT)
SUMV=0.0
DO 20 I=1, NT
SINA(I)=A(I)
COSA(I)=B(I)
20 SUMV=SUMV+B(I)*DEL8(I)*Y2(I)**2
SUMV=SUMV*5.9265

L=1
DO 30 I=1, NT
RB(I,L)=A(I)
30 RB(I,L+1)=B(I)
REWRITE 4
NSIG=NSIGA
CALL AXI
REWRITE 13

BLANK=0.0
READ (13) DUMMY
READ (13) DUMMY
WRITE (13) NN, BLANK, (X2(I), I=1, 199)
WRITE (13) NN, BLANK, (Y2(I), I=1, 199)
WRITE (13) BLANK, (T5(I), I=1, 199)
END
SUBROUTINE AXI

* COMPUTE AXI8YMMETRIC VELOCITY COMPONENTS AND PRINT

COMMON MGNR(8), ISWITCH, IPUNCH, ITERA, ITERMAX, IMACH, I8EPR, INY(8), IFŁAŁI
1GS, FLG05, MN, AP, ATT, REF1, SREF, X8EPND, NN, BDN, ND(2), NT, NSIGA, IPRINT, 1
BAMJET, PTJET, T7E), R, RSTAR
COMMON /CG/ X1(200), Y1(200), X2(200), Y2(200), DEL8(200), SINA(200), COAX
1SA(200), XP(200), YP(200)
COMMON /TC/ RB(200, 2), SIG(200, 1), A(200), B(200), Z(200), PHI(200, 1), XAXI
1N(200, 1), T(200, 1), TS(200, 1), NSIG, INP, NI, SUMV, SUMM(4)
COMMON /TMD/ NN, T1(200, 1), VY(200, 1), VT(200, 1), TH(200, 1), CP(200, 1), BAXI
1UMTDS(4)
COMMON /Tm/ NN, VX(200, 1), (VV, T), (VT, TS), (TH, SIG), (CP, T3)

EQUIVALENCE (VX, XN), (VV, T), (VT, TS), (TH, SIG), (CP, T3)
REAL MN
INTEGER FLG05, BDN

NCNT

NN=1
NP=NINP

* READ AXIS SIGMAS
SUMM(N)=0.0
SUMMTDS(N)=0.0
```
APPENDIX

READ (3) (B(I,N),I=1,NC)

DO 20 I=1,NT

READ (9) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)

DO 20 I=1,NT

READ (3) (B(I,N),I=1,NC)

* NO. OF MIDPOINTS LOOP

C

* READ MATRICES A,B,Z

C

READ (9) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)

* NO. OF FLOWS LOOP

N1=0
N1=N1+2
SN=0.0
ST=0.0
SP=0.0

* NO. OF ELEMENTS LOOP

DO 10 J=1,NT

SN=SN+A(J)*B(J)*Z(J)

10 CONTINUE

30 CONTINUE

IF (MN.EQ.0.0) GO TO 60

C * MACH NO. ADJUSTMENT

C

DO 30 I=1,NT

SN=SN+A(J)*B(J)*Z(J)

30 CONTINUE

IF (MN.EQ.0.0) GO TO 60

C * ELIMINATE MACH NO EFFECT FOR PRINTOUT

C

DO 40 I=1,NI

40 STOP
APPENDIX

60 CONTINUE
   IF (FLG400.EQ.0) RETURN

   * OFF-BODY POINT

   DO 80 J=1,NP
      READ (9) (A(J),J=1,NT),(B(J),J=1,NT),(Z(J),J=1,NT)

   * NO. OF FLOW

   SX=0.0
   SY=0.0
   SP=0.0

   * NO. OF ELEMENTS LOOP

   DO 70 J=1,NT
      SX=SX+A(J)*SIG(J,N)
      SY=SY+B(J)*SIG(J,N)
      SP=SP+Z(J)*SIG(J,N)

   70 CONTINUE

   IF (MN.EQ.0,0) 60 TO 110

   * MACH NO. ADJUSTMENT

   DO 90 I=1,NP
      BB=O.0

   90 CONTINUE

   DO 100 I=1,NP
      XP(I)=XP(I)*D3

   100 CONTINUE

   * LABRUJERE COMPRESSIBILITY CORRECTION

   IF (IMACH.GE.2) BB=1.0*MN**2*VX(I,N)
   VX(I,N)=VY(I,N)*D3/BB
   VX(I,N)=(VX(I,N)-1.0)/BB+1.

   80 CONTINUE

   DO 100 I=1,NP
      XP(I)=XP(I)*D3

   100 CONTINUE

   * COMPUTE VT AND THETA

   DO 120 I=1,NP
      VT(I,N)=SQRT(VX(I,N)**2+VY(I,N)**2)
   120 CONTINUE

   * PRINT AXIS FLOW (OFF-BODY) OUTPUT

   L=1
   I=1
   LCTR=45
   WRITE (6,170) HEDR
   WRITE (6,180)
   WRITE (6,190)

   170 CONTINUE
   CP2=(U.*D5*(1.-VT(I,L)**2))**S.5/5,0,)/D4
   XM2=VT(I,L)*MN/40.5*(VT(I,L)**2-1.)
   WRITE (6,210) I,XP(I),YP(I),VX(I,I),VY(I,I),VT(I,I),TH(I,I),XM2,CPAXI
   I=I+1
   IF (I.GT.NP) GO TO 150
   I=1,LE,LCTR) GO TO 140
   LCTR=LCTR+45
   GO TO 130
PROGRAM VICE FIV

C VISCOUS FLOW/POTENTIAL FLOW INTERFACE PROGRAM

COMMON HEDR(8),I8 witch,I 8 punch,IT ERA,ITERAX,IMACH,ISEP,INT(8),IFLAF
1G* . FIG 05,MO,APT,ATT,REL,RNP,NN,NO(2),NT,NSIGA,IPRINT,FIV
2AMJET,PTJET,TTJET,RTJET,RSTAR FIV

COMMON /SAVE/ VDUM(402),RD8(201),XIN,VDUM2(347)
DIMENSION X(200), R(200), CP(200), ME(200), THETA(200), CAPH(200)
1 CF(200), CAPH(200), CF(200), CD(200), CRPUNCH(200), FIV
2X0(200), RO(200), CS(200), R(200), UI(200), DELI(200), RET(200), FIV
3TAU5(200), PTT(200), FNM(200), DELT(200), FLOT(19), FLOT(7)
INTEGER FLG05,BDN
REAL MO,ME

C INITIALIZE

G=1.4
G1=(G-1.)/2.
G2=G/(G-1.)
G3=1./(G-1.)
G4=G+1./2.
G5=G+2*
G6=G/G1
TT=ATT
PT=APT
ME(1)=0.0
THETA(1)=0.0
CAPH(1)=1.3
CAPH(1)=1.3
CF(1)=0.0

C READ XO,RO,X,R,AND CP FROM TAPE13

REWIND 13
READ (13) XO,(XO(I),I=1,NXO)
READ (13) RO,(RO(I),I=1,NXO)
READ (13) NUM,(X(I),I=1,NM)
READ (13) NUM,(R(I),I=1,NM)
READ (13) CP

C OBTAIN CP AND R AT ORIGINAL X

NTAB=1
IORDER=2
IPT=1
DO 10 IM=2,NM
CALL IUNI (200,NUM,X+NTAB,CP,IORDER,XO(I),CS(I),IPT,IER)
10 CONTINUE
APPENDIX

```
DO 20 J=1,NUM
CP(I)=CS(I)
R(I)=RO(I)
Y(I)=Y0(I)
20 CONTINUE
REWIND 13
DO 30 J=1,NUM
IF (ITERA.EQ.0) RDS(J)=R(J)
X(J)=X(J)*REFL
R(J)=R(J)*REFL
RDS(J)=RDS(J)*REFL
30 CONTINUE
DO 40 I=1,NXO
XO(I)=XO(I)*REFL
RO(I)=RO(I)*REFL
40 CONTINUE

CALCULATE FREE STREAM CONDITIONS
PO=PT*M+G1*MO**2)**G2
QINF=G2/PO+MO**2
RG=286.96
P1=3.15926
CP(I)=(PT-PO)/(0.5*PO*6*MO**2)

CALCULATE PLUME BOUNDARY AND VELOCITY USING ONE DIMENSIONAL METHOD
DO 50 I=1,200
UI(I)=0.0
RI(I)=0.0
50 CONTINUE
IT=INT(6)
NJ=NUM/IT+1
IF (NJ.EQ.0) GO TO 60
RJET=HJET*REFL
K=0
DO 80 I=IT,NUM
K=K+1
PE=QINF*CP(I)*PO
PRAT=PTJET/PE
IF (I.OT.IT) GO TO 70
PCRIT=(1.0+G1*AMJET**2)**G2
IF (PRAT.GT.PCRIT) GO TO 60
RSTART=PRAT**(1.0/G)*SORT(G1**G6/G1*(1.0-PRAT**(*1.0/G2)))
RSTART=RSTART***(1.0/G)**G5
RSTART=RSTART***(1.0/G)**G6
GO TO 70
60 ASOA=G4**G5*XME/(1.0+G1*XME**2)**G5
RSTART=RSTART***(1.0/G)**G5
RSTART=RSTART***(1.0/G)**G6
70 XME=SORT((PRAT**(*1.0/G2)=1.0/G1)
ASOA=G4**G5*XME/(1.0+G1*XME**2)**G5
IF (I.OT.IT) R(I)=RSTART/SORT(A80A)
UI(K)=XME*SORT(G1**G7**G8**G9**G10**G11)
RI(K)=R(I)
80 CONTINUE
RJET=RJET/REFL
90 CONTINUE

PREPARE INPUT TO SUBROUTINE VISCOUS
INT(1)=NUM
INT(2)=1
INT(6)=ISEP=2
FL2(1)=12
FL2(2)=0
FL2(3)=PT/FL2(1)**2/47.880258
```
APPENDIX

```
FL0T(4) = IT = 1,8
FL0T(5) = MC
FL0T(6) = IT E R A + 1
FL0T(7) = MG
FL0T(8) = C A PH I (INT(2))
FL0T(9) = T H ETA (INT(2)) * FL0T(1)/3048
FL0T(10) = I P R I NT
FL0T(11) = U I (1) * FL0T(1)/3048
FL0T(12) = 0.0,25
FL0T(13) = 0,0
FL0T(14) = 0.0
FL0T(15) = 8 8 E P N D * R E F / F L0T(1)/3048
IF (1 T E R A .EQ .0) H I N X (N U M)
IF (1 N T(7) .GT .0) C A L L 8 M I N T (X, CP, N U M, IN T(3), N U M)
X I N = X I N / F L0T(1)/3048
D O 100 I = 1, N J
X(I) = X(I)*FL0T(l)/3048
R(I) = R(I)*FL0T(l)/3048
R D S (I) = R D S (I) * FL0T(1)/3048
C
C C H A N C E O U T P U T F R O M V I S C O U S T H M E T E R S
C
FL0T(7) = FL0T(7)/FL0T(1)/3048
X I N = X I N / F L0T(1)/3048
D O 120 I = 1, N U M
X(I) = X(I)*FL0T(1)/3048
R(I) = R(I)*FL0T(1)/3048
R D S (I) = R D S (I) / F L0T(1)/3048
R C P U N C H (I) = R C P U N C H (I) / F L0T(1)/3048
T H ETA (I) = T H ETA (I) / F L0T(1)/3048
D E L TA (I) = D E L TA (I) / F L0T(1)/3048
D E L T I (I) = D E L T I (I) / F L0T(1)/3048
C
C C A L C U L A T I O N O F D R A G C O E F F I C I E N T S
C
C D F (1) = 0,0
C D P (1) = 0.0
C D T (1) = 0.0
R O L D = 0.0
G O L D = 0.0
D O 140 J = 2, N U M
P E = P O *(1.0+ G/2.*) * M O**2 * C P (J)
N E W = G/2. * P E (J)**2
R N E W = R O (J)
A N G L E = T A N ((R N E W - R O L D )/(X (J) - X (J-1)))
S L E P I = (R N E W - R O L D ) * S Q R T ((R N E W - R O L D )**2 + (X(J) - X(J-1))**2)
C D T (J) = C D F (J) * C D P (J)
R O L D = R N E W
G O L D = N E W
C C C O N T I N U E
C
C C O U T P U T D A T A
```

APPENDIX

C
DO 150 N=1,NUM
X(N)=X(N)/REFL
R(N)=R(N)/REFL
THETA(N)=THETA(N)/REFL
DELTA(N)=DELTA(N)/REFL
RCPUNCH(N)=RCPUNCH(N)/REFL
RD8(N)=RD8(N)/REFL
DEL(N)=DEL(N)/REFL
CONTINUE
IF (ITERA.LT.IPRINT) GO TO 170
FLOTOTO(7)=FLOTOTO(7)/REFL
XINND=XIN/REFL
N1=1
150 CONTINUE
N2=N1+34
IF (N2.GE.NUM) N2=NUM
WRITE (6,200) HEDR,ITF,A,MO,TT,PT,REFL,8REF
WRITE (6,220) FIV
WRITE (6,230) (X(N),R(N),CP(N),CF(N),CDP(N),COF(N),CDT(N),RD8(N),RFIV
1CPUNCH(N),DELTA(N),OELI(N),THETA(N),CAPH(N),N*N1,N2)
TF (N2.GE.NUM) GO TO 170
N1=N2+1
GO TO 160
CONTINUE
REWIND 13
READ (13) BLANK
READ (13) BLANK
NINDEX=NUM
WRITE (13) NN,(X(I),I=1,NN)
WRITE (13) NN,(RO(I),I=1,NN)
WRITE (13) NN,(RCPUNCH(I),I=1,NN)
WRITE (13) NN,(CP(I),I=1,NN)
WRITE DATA ON TAPE 7 FOR RESTART
IF (IPUNCH.LT.1) GO TO 190
IF (ITERA.NE.ITERMAX) 80 TO 180
DO IB1=NXO
XO(I)=XO(I)/REFL
CONTINUE
REWIND 7
WRITE (7,270) HEDR
WRITE (7,240) IBWITCH,IPRINT,IPUNCH,IDM,IMACH,IBEP,(INT(I),I=1,IPRINT)
1.7,IFLAGS
WRITE (7,280) MO,APT,ATT,REFL,8REF,8SEPNL
WRITE (7,280) AMJET,PTJET,TTJET,RJET
WRITE (7,240) NN
WRITE (7,290) (XO(I),I=1,NN)
WRITE (7,300) (RO(I),I=1,NN)
WRITE (7,350) (RCPUNCH(I),I=1,NN)
WRITE (7,260) (CP(I),I=1,NN)
190 CONTINUE
FORMAT (1H1,BA10,5X,14MTERATION NO.,I2/2X,4HMO =,F7.4,4X,4HTT =FIV
1,F7.2,7H KELVIN,4X,4HPT =F10.1,8H PASCALS,4X,3HML =F10.6,7H METERFIV
28,X,6HSREF =F10.6,10H 80 METERS//)
210 FORMAT (4X,5HBOUNDARY LAYER SEPARATION AT X/L =,$F10.6,12X,36MBOUNFIV
10YAR REATTACHMENT AT X/L =,$F10.6,12X,36MBOUNFIV
220 FORMAT (5X,5X/L,5X,5H/L,5X,2HCP,6X,2HCP,6X,3HCP,5X,3HCP,5X,3HCPFIV
10T,4X,5HRO8/L,4X,4HRC/L,2X,6HMDEL/L,3X,5HDEL/L,2X,7HTHETA/L,3X,1HMRFIV
240 FORMAT (1X,13FB.4)
240 FORMAT (16IS)
APPENDIX

![Image of the document content]

250 FORMAT (6F10.6,6X,2HRC) FIV 14A
260 FORMAT (6F10.6,6X,2HCP) FIV 14B
270 FORMAT (6A10) FIV 290
280 FORMAT (F10.6,F10.1,F10.2,3F10.6) FIV 251
290 FORMAT (6F10.6,6X,2HCO) FIV 252
300 FORMAT (6F10.6,6X,2HNO) FIV 253
END FIV 254

SUBROUTINE VISCU8 (INT,FLOT,XA,RAD,CP,RJ,FLOT,RAO,A,THR,DEL51VIS 1
1,H51,CFA,DEL1,H1,RET,TAN51,PTB51,FNN51) VIS 2

C

C VISCOUS FLOW SUBROUTINE PACKAGE VIS 4

C COMMON /SAVE/ 88(201),SC(201),Y(201),XIN,XSEP(20),DELSV(20),YOUTVIS 6
1(201) VIS 7
DIMENSION INT(8),FLOT(15),FLOT(7),XI(201),RAD(201),U(201),RV(201) VIS 8
100C(201),A(201),THR(201),DEL51(201),H51(201),CFA(201),DELSV(201) VIS 9
101R(201),RET(201),TAW51(201),PTB51(201),FNN51(201),CP(201VIS 10
3),RI(201),U(201),VBLC(201),SS(201),S1(201),DISTAR(201),C8V(VIS 11
4201),CPCV(24) VIS 12

C

ANA=FLOT(6) VIS 13
TAN=ANA VIS 14
THR(1)=0 VIS 15
DELS1(1)=0 VIS 16
DELT(1)=0 VIS 17
H51(1)=0 VIS 18
RET(1)=0 VIS 19
TAW51(1)=0 VIS 20
PTR51(1)=0 VIS 21
FNN51(1)=0 VIS 22
NN=INT(1) VIS 23
NAZ=INT(2) VIS 24
NMIN=INT(3) VIS 25
NMAX=INT(4) VIS 26
TJET=INT(5) VIS 27
NEXT=INT(6) VIS 28
IS=INT(7) VIS 29
IPRESS=INT(8) VIS 30
Z=FLOT(1) VIS 31
TW=FLOT(2) VIS 32
PT=FLOT(3) VIS 33
TT=FLOT(4) VIS 34
AMIN=FLOT(5) VIS 35
GAM=FLOT(7) VIS 36
MIX=FLOT(8) VIS 37
THR=FLOT(9) VIS 38
CM=FLOT(12) VIS 39
X8IN=FLOT(15) VIS 40
IF (IANT.EQ.1) XIN=XA(NN) VIS 41
R=3.35 VIS 42
GG=32.174 VIS 43
PFREE=Pt*(1A*(GA=1A)*&AMIN*2)**(GA/(1A=GA)) VIS 44
IF (ANA.GT.1) GO TO 20 VIS 45
DO 10 I=1,NN VIS 46
10 Y(I)=RAD(I) VIS 47
1(I)=RAD(I) VIS 48
YSEP=0 VIS 49
20 CONTINUE VIS 50
DO 30 I=1,NN VIS 51
30 BB(I)=3.1416*Y(I)**2 VIS 52
C

CALCULATE VELOCITY FROM CP
C

DO 40 I=1,NN VIS 53
PL=5A*PFREE*AMIN*2*CP(I)*PFREE VIS 54
AML2&2 / (GA=1A)*((PL/PT)**((1A=GA)/GA)=1A) VIS 55
IF (AML2.LE.0.0) AML2=0.0000000001 VIS 56

38
APPENDIX

CALL SHAPEJ (SS, SS, XA, NN)

C NEWBL CONTROL CALCULATION OF BOUNDARY LAYER
CALL NEWBL (VBLC, XA, YB, NAZ, NN, TIN, ZET, ATAN, GA, U, 81, CFA, MIX, TV)

C FIX DETERMINES MIN. CP. THE MIN CP IS USED AS START LOCATION IN SEARCH FOR SEPARATION
CALL FIX (NMIN, NMAX, CP, MI)

C 8EPA DETERMINES SEPARATION PROPERTIES
80 IF (MI == NN) 60, 170, 170
60 MM=MM+1
CALL 8EPA (XA, RAD, CP, AMIN, CFA(MI), DELI(MM), THR(MM), RET(MM), CP(MI), MM)
80 CONTINUE

80 CONTINUE
IF (IAN == 1) GO TO 100
IF (IAN == 3) GO TO 90

IF (IAN <= 1) GO TO 90
IF (XSTN* = 0.0) GO TO 90
IF (XSEP*XSEP(V(IAN-1), EG, 0.) GO TO 90
XSEP = XA(NEXT = 1)
WRITE (6, 220) XSEP
90 CONTINUE

C CALCULATE SEPARATION POINT
IF (ANA, EQ, 1.) GO TO 160
IF (ANA, GT, 2.) GO TO 100
DELSV(2) = ABS(XSEP*XSEP = XSEP(V(I)) = XSEP(V(1))
XSEP = XSEP(V(1))
GO TO 160
100 CONTINUE
IF (ANA, EQ, 3.) GO TO 100
IF (ANA, GE, 5.) GO TO 150
AVEDEL = 0.
IAN = IAN + 1
DO 110 IBJ = 1, IAN
110 AVEDEL = AVEDEL + DELSV(IBJ)
AVEDEL = AVEDEL / IAN

AML = SORT(AML, 2)
TL = TT/(1. + (GA=1.) * 5*AML*2)
U(1) = SORT(2. / (GA=1.) * GR = GC * (TT/TL))

SHAPEJ CALCULATES 1ST DERIVATIVE OF CONTOUR
CALL SHAPEJ (SS, SS, XA, NN)
NEWBL CONTROL CALCULATION OF BOUNDARY LAYER
CALL NEWBL (VBLC, XA, YB, NAZ, NN, TIN, ZET, ATAN, GA, U, 81, CFA, MIX, TV)

FINE DETERMINES MIN. CP. THE MIN CP IS USED AS START LOCATION IN SEARCH FOR SEPARATION
CALL FIX (NMIN, NMAX, CP, MI)

8EPA DETERMINES SEPARATION PROPERTIES

IF (MI == NN) 60, 170, 170
60 MM = MM + 1
CALL 8EPA (XA, RAD, CP, AMIN, CFA(MI), DELI(MM), THR(MM), RET(MM), CP(MI), MM)
80 CONTINUE

IF (IAN == 1) GO TO 100
IF (IAN == 3) GO TO 90

40 IF (IAN, EQ, 1.) GO TO 160
IF (IAN, GT, 2.) GO TO 100
DELSV(2) = ABS(XSEP*XSEP = XSEP(V(I)) = XSEP(V(1))
XSEP = XSEP(V(1))
GO TO 160
100 CONTINUE
IF (IAN, EQ, 3.) GO TO 100
IF (IAN, GE, 5.) GO TO 150
AVEDEL = 0.
IAN = IAN + 1
DO 110 IBJ = 1, IAN
110 AVEDEL = AVEDEL + DELSV(IBJ)
AVEDEL = AVEDEL / IAN

AML = SORT(AML, 2)
TL = TT/(1. + (GA=1.) * 5*AML*2)
U(1) = SORT(2. / (GA=1.) * GR = GC * (TT/TL))

SHAPEJ CALCULATES 1ST DERIVATIVE OF CONTOUR
CALL SHAPEJ (SS, SS, XA, NN)
NEWBL CONTROL CALCULATION OF BOUNDARY LAYER
CALL NEWBL (VBLC, XA, YB, NAZ, NN, TIN, ZET, ATAN, GA, U, 81, CFA, MIX, TV)
IF (ABS(XSEP-XEPSV(IAN-1)) .LT. 2.*AVEDEL) GO TO 140
  120 XSEP=XEPSV(IAN-1)+2.*AVEDEL
  GO TO 140
  130 XSEP=XEPSV(IAN-1)+2.*AVEDEL
  CONTINUE
  XSEP=AEBIN(XEPSV(7),XEPSV(6),XEPSV(5))
  XEPSV(IAN)=XSEP
  CONTINUE
  DO 180 I=1,NN
  CSV(I)=CP(I)
  IF (XSIN.NE.0.) XSEP=XSIN
  IF (ANA.GT.4) GO TO 200
  C ZERO CP FOR THE FIRST 4 ITERATIONS
  DO 190 I=1,NN
  RATIO=0.
  190 CSV(I)=CP(I)*RATIO
  C XSEP CALCULATES THE AERODYNAMIC CONTOUR
  CALL XEP (NN,XA,RAD,CSV,XSEP,AEBIN,G,A,TT,PT,RADO,DSTAR,Y,ANA,IJET,NN)
  CALL POWER (X,N)
  CALL SUMA (NX,NY,LZ, X, CS, 1, CS2, CS3, SS, LS, X1)
  CALL SHAPEJ (SS, SS1, X, NN)
  RETURN
END SHAPEJ

SUBROUTINE POWER (X,N)
  DIMENSION X(1)
  COMMON /COEFFX X2(201),XJ(201),XJ(201)
 extrême(I),UJ(I),C
  CALL POWER (X,N)
  GO TO 10
  10 CONTINUE
  RETURN
END POWER

SUBROUTINE SUMA (NX,NY,LZ,X,8,C1,C2,C3,81,L)
  DIMENSION C188(201), C288(201), C388(201), SS(1), LS(1), X(1)
  CALL POWER (X,N)
  CALL SUMA (2,NN=1,3,X,8,C188,C288,C388,81,1)
  S1(1)=0.0
  S1(NN)=0.0
  RETURN
END SUMA

DIMENSION X(1)
COMMON /COEFF/ X2(201),XJ(201),XJ(201)
DO 10 I=1,NN
  X2(I)=X(I)*X(I)
  X3(I)=X2(I)*X(I)
  X4(I)=X3(I)*X(I)
  CONTINUE
  RETURN
END

DIMENSION X(1)
DO 20 I=1,NN
  X2(I)=X(I)*X(I)
  X3(I)=X2(I)*X(I)
  X4(I)=X3(I)*X(I)
  CONTINUE
  RETURN
END

SUBROUTINE SUMA (NX,NY,LZ,X,8,C1,C2,C3,81,L)
APPENDIX

THIS SUBROUTINE CURVES FITS A PARABOLIC ARC THRU LEAST SQUARES

COMMON /COEFF/ X2(201), X3(201), X4(201)
DIMENSION X(1), B1(1), C1(1), C2(1), C3(1)
DOUBLE PRECISION SUM1, SUM2, SUM3, SUM4, SUM5, SUM6, SUM7

LN=LZ/2
C1(NX-1)=.0, 0
C2(NX-1)=.0, 0
C3(NX-1)=.0, 0
DO 30 J=NX, 1, -1
SUM1=0, 0
SUM2=0, 0
SUM3=0, 0
SUM4=0, 0
SUM5=0, 0
SUM6=0, 0
SUM7=0, 0
M=J-LN
MM=J+LN
DO 10 I=M, MM
SUM1=SUM1+X(I)
SUM2=SUM2+X2(I)
SUM3=SUM3+X3(I)
SUM4=SUM4+X4(I)
SUM5=SUM5+(X(I)*B(I))
SUM6=SUM6+(X2(I)*B(I))
SUM7=SUM7+B(I)
CONTINUE

AC=SUM7
AD=SUM1
AC=SUM2
AD8SUM5/8UM1
AB8SUM2/8UM1
AF8SUM6/8UM2
AG=SUM3/8UM1
AM8SUM4/8UM2
A8=SUM3/8UM2
AAR=AA/LZ
ABAAR=AD
CA=AAAR=AF
ABAR=BLZ
R=ABAR=AE
D=ABAR=AI
ACAR=AC/LZ
E=ACAR=AG
G=ACAR=AH
RM=1D=10
DM=1D=10
EB=E/B
AM=EB=G/D
IF (ABS(AH), LE, 0, 1D=10) GO TO 20

AR=LA/B
C3(J)=A8B=C/D/AM
C2(J)=A8B=EB+C3(J)
C1(J)=AKAR=C2(J)*ABAR=C3(J)*ACAR
GO TO 30

CONTINUE

C3(J)=C3(J+1)
C2(J)=C2(J+1)
C1(J)=C1(J+1)

CONTINUE

IF (L.EQ.0) RETURN

COMPUTE 1ST DERIV. OF X VS B CURVE.
APPENDIX

```fortran
DO 40 J=NX, NY
  S1(J)=C2(J)+2*C3(J)*X(J)
  CONTINUE
DO 50 J=1, LN
  K=NX-LN+J-1
  S1(K)=C2(NX)+2*C3(NX)*X(K)
  CONTINUE
DO 50 J=1, NY
  S1(I)=C2(NY)+2*C3(NY)*X(I)
  CONTINUE
RETURN
END

SUBROUTINE NEWBL (VBLC,X,Y0,FX,THR,AM,DEL1,RET,THR,D8TA,THR,DEL51,H51,TAW51,FNN51,H,DRAG)
  DIMENSION H1(1), D8TAR(201), U(1), VBLC(1), X(1), Y0(1), S(1), S1(NBL 11), CFA(201), AM(1), THR(1), DEL1(1), RET(1), DEL51(1), H51(1), TAW51(5)
  COMMON /SAVE/ 88(201), 8C{201, NBJBNN.NAZ+1

DO 10 I=1, NNAZ
   VBLC(I)=U(I)
10 CONTINUE
DO 20 KJ=1, NBJ
   NJA=NAZ+KJ-1
   CPR(KJ)=VBLC(NJA)
   YBAR(KJ)=X(NJA)
   BC(KJ)=Y0(NJA)
20 CONTINUE
  CALL BLC (PT, TT, YBAR, RC, CPB, TW, Z, NB, DBST, THR, H, H1, CFA, AM, GA, NBL 1)
  INDL, RET, THR, DEL51, H51, TAW51, PFR51, FNN51, DRAG)
  PT=ABC
  IF (NAZ) 60,60,70
30 DO 40 NJ=1, NBJ
   NAZ=NAZ+NBJ
   CPR(NAJ)=CFA(NJ)
   BC(NAJ)=AM(NJ)
   YBAR(NAJ)=DBST(NJ)
40 CONTINUE
  DO 50 NJ=1, NBJ
   YBAR(NJ)=D8TAR(NJ)
50 CONTINUE
  DO 100 I=1, NNAZ
   CFA(I) =CPB(I)
   AM(I) =BC(I)
   D8TAR(I)=YBAR(I)
   IF (8(I)-.1E-8) 70,70,70
70 RCDO=.0
   GO TO 90
80 CONTINUE
RCO=S1(I)/(2.0*8QRT(3.1416*8(I)))
RCO=ABR(RCO)
90 CONTINUE
DE=RCO**2+1.0
SUG=1.0/DE
AMB=SQRT(SUG)
D8TAR(I)=D8TAR(I)/AMB
CONTINUE
IF (ANA,GT,1.) GO TO 120
DO 100 I=1, NNAZ
   SC(I)=.0
100 S8R(I)=.0
120 ABC=.0
```

42
APPENDIX

DO 140 I=1,NN
IF (ANA.LE.3.) GO TO 130
DBSTAR(I)=25*DBSTAR(I)+5*BB(I)+25*BC(I)
130 SC(I)=BB(I)
140 CONTINUE
RETURN
END

SUBROUTINE BLC (PT,TT,VX,VY,TW,W,Z,NN,DBSTAR,THRR,MIX,HICH,CFA,AM,BLC 1
GAM,DELI,RET,TIB,DEL5,NN5,RET5,NN55,DRAG)

DIMENSION AM(1), VX(1), VY(1), DBSTAR(1), HICH(1), DEL1(1), CBL
IF (ANA.LE.3.) GO TO 130
DBSTAR(I)=25*DBSTAR(I)+5*BB(I)+25*BC(I)
130 SC(I)=BB(I)
140 CONTINUE
RETURN
END

TF(X)=1.7*X**2
PF(X)=TF(X)**3.5
TANE(X)=1.+.178*X**2
H2XX=X*(X-1)**2/2.
H3XX=2.*HXX/HX-1.)*HXX-0.5*C(HX-1.)*.3/4.3
G1=GAM/(GAM-1.)**2.
G2=GAM/(GAM-1.)
DBSTAR(1)=MIX*THRR
THRR=THRR
IF (THRR.LT.0.0001) THRR=0.0001
DO 10 I=1,NN
X(I)=VX(I)/Z
AM(I)=V(I)/TT**.5*QRT(TT**2.)/32./3.17)
10 IF (AM(I).LE.0.00001) AM(I)=AM(2)
PT=PT*Z*Z
L=1
HIF=1.3
U=2.7E-08*TT**2.1.5/(TT**4.98/776.)/32./17)
IF (HIX) 20,20,30
20 HX=1.3
GO TO 40
25 IF (HIX) 20,20,30
30 HX=MIX
40 DO 230 I=1,M
DM=AM(I+1)-AM(I)
DYVY(I+1)=VY(I)
DXVX(I+1)=X(I)
XXNN=NN
DLMM=AB(DM/AM(I)*DM/AM(I+1)+0.001)*DX/THR*Z*XXNN*MD
IF (Y(I+1)) 50,60,50
50 IF (Y(I)) 50,60,50
60 IF (N=10) 70,70,80
70 N=10
80 IF (J+N) 90,100,100
90 N=30
100 N=1
DX=DX/5
YYVY(I)=DY/2./5
DY=DY/8
AAM=AM(I)*DM/2./8
DM=DM/8
DL=8QRT(DX**2.+DY**2)
APPENDIX

```
N=8
DO 220 J=1,N
YY=YY+DY
AA=AA+DM
TE=TE+TF(AA)
TA=TE+TAE(AA)
IF (THW) 110,110,120
110
TH=TAW
GO TO 130
120
TH=THW
130
TH=(TA+TE)/2.++,22*TE=(TAE(AA)=1.0)
TH=TH+(TH=THW)/2.
THP=THTR
THTR=(THM+THT/2.)
H=HD/2.+-HM
A=1.25*EXP(-1.561*HI)*(AA+AA)*((=-268)*(TE/TR)**.72*(TE/TT)**3)
1.268
AAA=(TT/TR)**0.609
THM=AA+DL=THM+DM/AA(2.++HI*(TH/TT=1.)**HIF/HM)
THT=(THR++DY/YY)*THT)
THTR=(THM+THT/2.)
HD=DM/AA+2.++(HI)**(H=E=1.0)(TH/TT=1.)**HIF/HM)
H=H+(HI=1.0)**.366*EXP(2.9*(HI=1.)**1./HI))
H=H+(HI=1.0)*EXP(2.9=(HI=1.)**1.)
HD**HIF/HM=THTR=AA+DL
IF (ABS(HD)/HI=2.) 150,150,140
140
HD=2.**HDD/ABB(HD)/HI
150
H=HMH+HD
THTR=(THM+THT)
IF (HI) 160,160,170
160
HI=5
HD=.0
GO TO 190
170
CONT=2.0
IF (HI=CONT) 190,190,180
180
HI=CONT
HM=CONT
HD=.0
190
TFAT=TA
THETHT=THTR=TFAT=3
CFB2=AA-**TFAT=3
IF (J+I=2) 200,200,210
200
RV=PT/PP(AA)**GRT(1.4/TE/1716.)**AA
210
HEAD=0.70+(AA+AA)**PT
HEAD=HEAD/((1.0+0.2*(AA+AA))**3.5)
CF2=CFCF+HEAD
DRA=DRA*6.2632=CFQ+YY+DX
RV=PT/PP(AA)**GRT(1.4/TE/1716.)**AA
THETHT=(TH/TT=1.)**HIF/HM
TH=THF+TFAT=2.++AA**2
U=2.27=E=0.8*TE=1.5/(TE=198.6)
RESRV=THTR/U
TH=THR=Z
DEL=THRH
DSTA=DEL(1.0)*DEL
220
CONTINUE
FNN=2.0/(HI=1.0)
FM=1.0+G10*(AM(I+1)**2)
F2=FM=1.0
FM=GAME*(AM(I+1)**2)/2.
THETHR/FM=3
DDD=TFAT=1.0*(FNN)/(2.0+FNN)/FNN
DSTA=DEL/FM=3
ABB=TFAT=88
```

44
APPENDIX

```fortran
DEL1(I+1) = 3000 * 4 + 5 * GAM + DEL1(I+1)
PTBARFM1 = G2 * (1.0 + FM3 * (1.0 + (DEL + THR) / DEL1(I+1)))
L+1
DEL51(L) = DEL
HS1(L) = H
THS1(L) = TAN
PTBS1(L) = PTBAR
FNNS1(L) = FN
CF1(I+1) = CF
MICM(I+1) = MIC
RET(I+1) = RB
THR(I+1) = THR
CONTINUE
CFA(I+1) = CFA(I)
RETURN
END
SUBROUTINE FIX (NMIN, NN, CP, MI) FIX 1
C
DIMENSION CP(1) FIX 3
C
MIMNMIN FIX 5
PMIN = CP(NMIN) FIX 7
DO 20 I = NMIN, NN FIX 9
10 PMIN = CP(I) FIX 11
20 CONTINUE FIX 13
RETURN
END
SUBROUTINE SEP* (X,R,C
P,E*I»CFl»DEL 1»TH£T*1,R£TMJ,CP1,N8N,NEN,CPRT8PA 1
1)
C
DIMENSION X(201), CP(201), CPDC201), C1C20J), C2(201), C3(8PA U
1201), CPRT(U), YY(201)» P3PK201), E(201) SPA 5
COMMON /COEFF/ X2(201),X3(201),XUC201) SPA 6
COMMON /BCB/ TTRAT(201),PTRAT(201),PTRN8(201),URAT(201),EM(201),WR8PA 7
1(201),PHIR(201),TNTTE,GMMA,GMRAE,PMON,VMVE1,C,DDO,DSSD,BK,EDE1,SHPA 8
2FAC,8IG1,SIGMA1,SICS1 SPA 9
C
ASIN(X) = ATAN(X/SORTCl-X»xn SPA 11
C
EM1+0.0 SPA 13
BTAIL=0.0 SPA 14
ELLAT=0.0 SPA 15
ENTMAS=0.0 SPA 16
ENPRES=3.0 SPA 17
CF1CF1=1.0 SPA 18
GAMMA=1.4 SPA 19
GAMMA=GMMA SPA 20
EM1 = SQRT(5.0 * (1.0,2*EM1*EMI) * (1.0,7*EM1*EMI*CP1)* (1.0,GAM) / GAM) = 1BA 21
1.
EM1 = EM1
EM1 = EM1 / EMI
CP8G = CP1 + 200.0 * CF1 * (1.0, (GAM/2.0)*EMI*EMI*CP1)*EMI*EMI = EM1
CALCULATE CP(SEP) AND P(SEP) / PI USING GOLD8MHEID METHOD SPA 26
C
PB8PIG=1.45*5*GAM*CP8G*EMI*EMI SPA 29
PB8PIG = PB8PIG/(1.45*5*GAM*CP1*EMI*EMI)
C
STRATFORDS SEPARATION CALCULATION SPA 32
C
NX=84N+2 SPA 34
NY=84N+2 SPA 35
LZ=5 SPA 36
L=1 SPA 37
```

45
CALL SUMA (NX, NY, LZ, XCP, C1, C2, C3, CPDL)
RE = RETH1 * (X(NSN)/THETA1)
S = DEL1 * (3*RE**2)
RHS = 39 * (10**(4.6) * RE**(-1))
DIF = 0
JK = 0
GO 100 IF (NX) 90, 90, 10
10 CONTINUE
MLB = CP(1)*(8*CPD(I)**5)
DIF = RHS = MLB
IF (JK = 1) 20, 40, 40
20 CONTINUE
JK = 1
LNZ = 0
IF (DIF) 30, 30, 40
30 LNZ = 1
40 CONTINUE
IF (LNZ) 50, 50, 70
50 IF (DIF) 60, 60, 90
60 IAM = 1
GO TO 110
70 IF (DIF) 90, 80, 80
80 IAM = 1
GO TO 110
90 CONTINUE
GO TO 120
100 CONTINUE
GO TO 120
110 CONTINUE
CP8S = CP(I = 1)
CPRT(4) = CP8S
120 CONTINUE
C CALCULATE CPSEP AND PSEP/PI USING MODIFIED PAGE METHOD
C CP8P = CP1 + 0.38*(1 + (GAM2)*EM1*EM1*CP1)*EM1*EM1*EM1
C EVALUATE PROFILE PARAMETERS AT STATION 1
C
BK = 0.4
EM1 = 1.
TNT = T12
SIGMA1 = 1.2*EM1*EM1
SIGMA1 = BIG1/(1 + SIGMA1)
BIG1 = SQRT(SIGMA1)
SIGMAA = (SIGMA1)/(BIG1)
SIGMA = (SIGMA1)/(SIGMA1)**1.76
UTUE = SQRT((CF1/2)**2*(SIGMA1)/(1 + SIGMA1))**ASIN(SQRT(SIGMA1))
REDEL1 = RETH1*DEL1/THETA1
PX = 1.5*(1 + UTUE + ((1/BK)*ALOG(REDEL1*ABS(UTUE1)*SIGMA1)*UTUE1))
1)
1)
C DETERMINE B.L. PROFILE PROPERTIES AT STATION 1
C CALL PRPL (UTUE1, PX, EM1, 1.0, 2, YY)
C DETERMINE UPSTREAM BOUNDARY LAYER INTEGRAL PROPERTIES
C CALL FLUX (101, YY, EM1)
RETH1 = REDEL1*DD8D
AM = 881 + BLMN
AM = 1 + BLMN
C START SOLUTION PROCEDURE, ASSUME P8/P1
46
APPENDIX

DO DDOD=DDSD
DO DDOD=DDSD
UTUSSS=0.
PSPI(1)=PSPIG
PIPT=(1.0+2.0*EMI*EMI)*(GAM/(1.0+GAM))
PHSM1=PSPI(1)*PIPT
PHSM2=PIPT
EM8=EM1*8QRT(PHSM1/PHSM2)
PS8=0.5
EM1=1.0
GO TO 140

130
J=-1
IF (J.EQ.80) GO TO 290
PSPI(J)=PSPI(J=1)=0.1
PHSM1=PSPI(J)*PIPT
PHSM2=PIPT
EM8=EM1*8QRT(PHSM1/PHSM2)
PS8=0.5

C ENTRAINMENT AND FRICTION CONSTANTS
C
C ENSURE ENTRAINMENT FROM GREEN'S THEOREM
C
140
ELDEL1=ELBBT*BTAI/DEL1
APIPI=1.0+7.0*EMI*EMI*CP1
PGX=(1.0+7.0*EMI*EMI*(CP(N8N=1)))/APIPI
IJR=NEN
DO 230 II=NBN,NEN
PGK=PGX
PGX=(1.0+7.0*EMI*EMI*(APIPI))/IJR
IF (II=NBN) 150,150,160
PGX=PGX
GO TO 230

160
CONTINUE
DPG=PGX=PSPI(J)
IF (DPG) 190,190,170
170
IF (PGX=PSPI(J)) 180,220,220
180
DBG=PSPI(J)=PGX
DTG=ABS(PGX=PGX)
GO TO 210

190
IF (PGX=PSPI(J)) 220,220,220
200
DBG=PSPI(J)=PGX
DTG=ABS(PGX)
210
AL=X(I+1)-X(NBN)+(DBG/DTG)*(X(II)-X(II+1))
IJR=I+1
GO TO 240

220
CONTINUE
AL=X(NEN)-X(NBN)
240
ELDEL1=AL/DEL1
IF (J.LE.2) GO TO 260
TOTAL=0.
DO 250 II=NBN,NEN
150
TOTAL=TOTAL+CP(II)
CPAVE=TOTAL/(IJB=NBN+1)
CPVC=(CPV(1)*(1.0+7.0*EMI*EMI*CP1)=1.0)/(7.0*EMI*EMI)
ENPRE=(CPVC*CP1)/(CPAVE=CP1)
250
CONTINUE
FENT=1.0-DDOD/DDOD=3
FENT2=FENT+0.6169
FENT3=1.0
MENBL=ELDEL1,0299,FENT2,FENT3
GO TO 280
270
MENBL=ENTMA8
CONTINUE
CALL PRFL (UTUES3, PX3, EMS, P8P1(J), 1, YY)
CALL FLUX (101, YY, EMS)
AMAS3 = BLMN
AMOB8 = BLMON
ALH81 = \( (1.0 + P8P1(J)) \times (1.0 + AMEMBL) \times EM1 / EMS \)
ALH82 = \( (1.0 + 2.0 \times EM1 \times EM1) / (1.0 + 2.0 \times EMS \times EMS) \)
ALH83 = AMAS3 / AMAS3
ALH8 = ALH81 * ALH82 * ALH83
RHS1 = \( (1.0 / ENPRE8) \times (P8P1(J) - 1.0) \)
RHS2 = \( 1.4 \times EM1 \times EM1 \times AMOM1 \)
RHS3 = \( 1.4 \times AMEMBL \times EM1 \times EM1 \times AMOB8 \)
RHS4 = \( 5.0 \times CF1 \times CFHCF1 \times 70.0 \times EM1 \times EM1 \times ELDEL1 \)
RHS5 = \( 1.0 \times (1.0 / ENPRE8) \times (P8P1(J) - 1.0) \times P8P1(J) \)
RHS6 = \( P8P1(J) \times 1.4 \times EMS \times EM8 \times AMOB8 \)
RHS7 = \( RHS1 = RHS2 = RHS3 + RHS4 \) / (RHS5 = RHS6)
E(J) = RHS = ALH8
TEST = ABS(E(J))
IF (TEST.Le 0.00001) GO TO 300
IF (J, LE, 60) GO TO 130
IF (ABS(E(J) - E(J-1)) \( \times 0.001 \times AB8(E(J) + E(J-1)) \times \) .5) GO TO 300
BLNPEM(E(J) - E(J-1)) / (P8P1(J) - P8P1(J-1))
P8P1(J+1) = P8P1(J) + E(J) / SLOPE
IF (P8P1(J+1), LT, 0.0) GO TO 130
EM81 = \( (P8P1(J+1) \times P8P1) \times (1.0 + GAM) / GAM \) = 1.0
IF (EM81, LE, 0.0) GO TO 130
EM8 = EM1 \times SORF(EM81 / EM82)
J = J + 1
IF (J, EQ, 80) GO TO 290
GO TO 140
CONTINUE
GO TO 310
SOLUTION OBTAINED
DETERMINE B.L. PROPERTIES AT SEPARATION
DETERMINE DOWNSTREAM B.L. INTEGRAL PROPERTIES
CALL PRFL (UTUES3, PX3, EMS, P8P1(J), 2, YY)
DETERMINE DEL3/DEL1 AND SEPARATION PRESSURES
P8P1F = P8P1(J)
CPCV = \( (1.0 + 7.0 \times EM1 \times EM1 \times CP1) - 1.0 \) / \( (7.0 \times EM1 \times EM1) \)
RESULTS FROM CONTROL VOLUME APPROACH
CPRT(1) = CPCV
RESULTS FROM GOLDSCHMIEID
CPRT(2) = CPSG
RESULTS FROM MODIFIED PAGE METHOD
CPRT(3) = CPSG
CONTINUE
RETURN
END
SUBROUTINE PRFL (UTUES3, PX3, EMS, PKP1, IOPT, YY)
SUBROUTINE TO CALCULATE DISTRIBUTIONS OF PROPERTIES
DIMENSION YY(201)
APPENDIX

COMMON /BCBX,TTRAT(201),PTRAT(201),PTRNS(201),URAT(201),EM(201),WRPRF
1(201),PHIR(201),TWTTE,GAMMA,BLMN,BLMON,VWVE1,C,DSD,DDSD,BK,EME1,8HPRF
2FAC,SIG1,SIGMA1,BIG81

C
PI=3.1415927
EXP2=GAMMA/(GAMMA=1.)
EXP3=1./(GAMMA=1.)
GAM1=(GAMMA=1.)^2
GAM2=GAMMA+1.
GAM3=GAMMA=1.
G1=EM*EM
SIGMA=GAM1*G1/(1.+GAM1*G1)
SIGG1=SQRT(SIGMA)
SIGG2=1./SIGG1
SIGG3=ATAN(SIGG1/SQRT(1.+SIGG1*SIGG1))
URAT(1)=0.
TTRAT(1)=TWTTE
EM(1)=0.
YY(1)=0.
PTRAT(1)=(1.)/(1.+GAM1*EME1*EME1))**EXP2*PKPl
PTRNS(1)=PTRAT(1)
DO 40 I=2,101
AI=I-1
YY(I)=AI/100.
URAT(I)=SIGG2*SIGG3*SIGG1*(1.+GAM2/2.)/(1./BK)UTUE8TPR
1*BIG83*ALOG(YY(I)))
TTRAT(I)=TWTTE*(1.+.TWTTE)*ABS(URAT(I))
U2=URAT(I)*URAT(I)
EM(I)=SIGG3*SIGG1*SIGG3*(1.+GAM2/2.)**EXP2*(GAM2/2.)**EXP2*PKPl
CONTINUE
END
SUBROUTINE FLUX (K,Y,EME)
C SUBROUTINE TO CALCULATE MASS AND MOMENTUM FLUX OF BS,,
C ALSO CALCULATES DISPLACEMENT AND MOMENTUM THICKNESSES
DIMENSION Y(201), YYC201), BLMR(201), BLMOR(201)
COMMON /BCBX TTRAT(201),PTRATf201),PTRNS(201),URAT(201),EM(201)»WRPRF
lf20l)»PHIR(201)»TWTTE,6AMMA,BLMN,BLMON,VWVE1,C,D8D,DD8D»BK,EME1,8HPRF
2FAC,SIG1,SIGMA1,BIG81
DO 10 I=1,K
PRAT1=
TOTE1=(GAMMA=1.)*EM(I)*EM(I)/2.
TOTE2=(GAMMA=1.)*EME=EME/2.
TOTE=TOTE1+TOTE2
PRAT=TOTE/TOT
RLH=PRAT/TOTE
RLMR=RLHRA=URAT(I)
BLMR=BLMR*URAT(I)
IF (URAT(I).LE.O.) PRAT=0
CONTINUE
END
SUBROUTINE INTEG (I,YY,BLMR,AREA1)
C CALCULATION OF TOTAL PArSSURE DOWNSTREAM OF NORMAL SHOCK
C
10 PTRAT(I)=(1.)/(1.+GAM1*EME1*EME1))**EXP2*PKPl
IF (EM(I).LE.1.) 20,20,30
20 PTRNS(I)=PTRAT(I)
GO TO 20
30 PTRNS(I)=(GAM2*EM(I)*EME1*EME1)/2.)/(1.+GAM1*EME1*EME1))**EXP2*(GAM2/2.)**EXP2*PKPl
CONTINUE
END
SUBROUTINE FLUX (K,Y,EME)
C SUBROUTINE TO CALCULATE MASS AND MOMENTUM FLUX OF BS,,
C ALSO CALCULATES DISPLACEMENT AND MOMENTUM THICKNESSES
DIMENSION Y(201), YYC201), BLMR(201), BLMOR(201)
COMMON /BCBX TTRAT(201),PTRATf201),PTRNS(201),URAT(201),EM(201)»WRPRF
lf20l)»PHIR(201)»TWTTE,6AMMA,BLMN,BLMON,VWVE1,C,D8D,DD8D»BK,EME1,8HPRF
2FAC,SIG1,SIGMA1,BIG81
DO 10 I=1,K
PRAT1=
TOTE1=(GAMMA=1.)*EM(I)*EM(I)/2.
TOTE2=(GAMMA=1.)*EME=EME/2.
TOTE=TOTE1+TOTE2
PRAT=TOTE/TOT
RLH=PRAT/TOTE
RLMR=RLHRA=URAT(I)
BLMR=BLMR*URAT(I)
IF (URAT(I).LE.O.) PRAT=0
CONTINUE
END
SUBROUTINE INTEG (I,YY,BLMR,AREA1)
APPENDIX

CALL INTEG (I,YY,BLMOR,AREA2)
WR(I)=AREA1
PHiR(I)=AREA2
20 CONTINUE
BLMN=AREA1
BLMON=AREA2
DBD1=BLMN
DBD2=BLMON
SMAC=DBD1/DBD2
RETURN
END

SUBROUTINE INTEG (K,Y,Z,AREA)
C INTEGRATION USING SIMPSON'S RULE
C
DIMENSION Y(201),Z(201)
C
IF (K.GE.5) GO TO 10
IF (K.EQ.1) GO TO 80
IF (K.EQ.2) GO TO 90
IF (K.EQ.3) GO TO 100
IF (K.EQ.4) GO TO 110
10 AK=K
BK=AK/2.
KK=AK
CK=KK
IF (BK.CK) 30,20,50
C
40 N*K<20
J=0
B=0.
IF (J.GE.10) GO TO 50
EVEN=J.
EVEN=J+1
ODD=ODD+Z(1+J)
CONTINUE
50 AREA=(Y(2)-Y(I))/2.*(Z(1)+Z(N)+4.*(EVEN+Z(N+1))+2.*ODD)
IF (BK.CK) 70,60,70
C
60 AREA=AREA+(Y(K)=Y(K-1))*Z(K)+Z(K+1))/2.
RETURN
C
70 RETURN
50 AREA=AREA+(Y(2)-Y(I))/2.*Z(2)+Z(1))/2.
RETURN
90 AREA=AREA+(Y(2)-Y(I))/2.*Z(3)+Z(1))/3.
RETURN
100 AREA=AREA+(Y(2)-Y(I))/2.*Z(4)+Z(2)+Z(1))/3.
RETURN
110 AREA=AREA+(Y(2)-Y(I))/2.*Z(5)+Z(3)+4.*Z(2)+Z(1))/3.
RETURN
END

SUBROUTINE SEP (NN,XA,RAD,CP,XSEP,AMIN,GAMMA,TTO,PT,RADO,OBSTAR,Y,A9EP,1
1NA,IJET,NEXT,RI,UIJ,C)
C
APPENDIX

DIMENSION DSTAR(1), RADO(1), XA(1), RAD(1), CP(1), Y(1), X834(201)SEP 4
1, Y834(201), RI(1), UJ(1) SEP 5
COMMON /SAVE/ SB(201), SC(201), YJB(201), XIN, XSEP, V(20), DELB(20), YSEP 6
1OUT(201) SEP 7
C SEP 8
10 Y(I)=RADO(I) SEP 9
IF (XSEP.GT.X0) GO TO 30 SEP 10
DO 20 I=1,NN SEP 11
20 RADO(I)=Y(I)+DSTAR(I) SEP 12
GO TO 120 SEP 13
30 CONTINUE SEP 14
DO 40 I=1,NN SEP 15
40 Y(I)=XSEP SEP 16
IF (XSEP.GT.XA(I)) 50,40,40 SEP 17
90 CONTINUE SEP 18
50 RTAN(Y(IS+1)=Y(IS))/XA(IS)=XA(IS+1) SEP 19
RTAN=ATAN(RTAN) SEP 20
YSEP=Y(IS)+(XSEP-XA(IS))-(Y(IS-1)=Y(IS))/(XA(IS-1)-XA(IS))) SEP 21
INC=1 SEP 22
X834(IS)=XSEP SEP 23
Y834(IS)=YSEP SEP 24
DO 60 I=1,NN SEP 25
60 Y(IS)=X(A(I)) SEP 26
RDEG=180./3.1415926 SEP 27
RTAN=RTAN*RADUSEG SEP 28
NEXT=IS+2 SEP 29
JJJB=0 SEP 30
IF (ANA.GE.1.) JJJB=JJJB+1 SEP 31
IF (ANA.GE.9.) GO TO 70 SEP 32
CALL B834 (IC,AMIN,GAMMA,TTO,RTAN,X834,Y834,CP(IS=1),YOUT,JJJB,YSEP 33
NEIN,RJ,UIJ,C,ANA) SEP 34
70 CONTINUE SEP 35
IF (ANA.EQ.1.) GO TO 100 SEP 36
70JB=2 SEP 37
DO 90 I=1,NN SEP 38
90 Y(I)=YOUT(JB) SEP 39
RADO(I)=YOUT(JB)+DSTAR(I) SEP 40
IF (Y(I).LT.Y(I-1)) GO TO 80 SEP 41
IF (RADO(I).LT.RADO(I-1)+Y(I)=Y(I-1)) RADO(I)=RADO(I-1)+Y(I)=Y(I-1) SEP 42
GO TO 90 SEP 43
80 IF (RADO(I).LT.RADO(I-1)) RADO(I)=RADO(I-1) SEP 44
90 JB=JB+1 SEP 45
GO TO 120 SEP 46
100 CONTINUE SEP 47
JB=2 SEP 48
API=2.*Y(IS)+DSTAR(IS)+DSTAR(IS)**2 SEP 49
DO 110 I=1,NN SEP 50
110 Y(I)=YOUT(JB) SEP 51
ARGF=4.0*Y(I)**2+4.0*API SEP 52
DSTAR(IS)=2.*Y(I)+SQRT(ARGF))/2.0 SEP 53
RAD0(I)=YOUT(JB)+DSTAR(I) SEP 54
IF (RAD0(I).LT.RADO(I-1)) RADO(I)=RADO(I-1) SEP 55
JB=JB+1 SEP 56
120 CONTINUE SEP 57
C SEP 58
C SEP 59
110 C SEP 60
RETURN SEP 61
C SEP 62
END SEP 63
SUBROUTINE B8S4 (NBT,FMS,GAMMA,TTO,PT,ABOD,XL,RAD,CPIN,YSTR,IJET,NBEX
1EXT,RI,UJ,C,ANA)

C AXI-SYMMETRIC SEPARATION ANGLE PROGRAM

C COMMON /SAVE/ SB(201),SC(201),Y(201),XIN8
DIMENSION XSTR(201),YSTR(201),M1V(201),HV(201),UHEV(201),P8B8S 7
IIV(201),PIV(201),AMIV(201),XJET(201),CPIN(1),R1(1),UJ(1),X88S
2L(201),RAD(201)

C
N8EP=1
NPT=101
EPSLB=0.00001
DJDXM=0.05
DUMDX=4000
DELLOC=0
IL8V=0
SLOPL=0
DO 10 IL=1,20
10 HV(20)=0
DEGRAD=1415926/180.
DO 20 IL=1,NST
XSTR(20)=XL(IL)
20 YSTR(20)=RAD(IL)
ABOD=ABOD+DEGRAD
ABODSV=ABOD
AT=SQRT(GAMMA*32.17453,15+TTO)
ITDP=0
PB=PT*(1.+GAMMA)=2.**(GAMMA/1.+GAMMA))
IF (NB8P,LE,0) N8EP=1
IL=0
IUE=0
P8L=0
P8B=0
DO 30 I=N8EP,NB8T
PIV(20)=PB*(1.+GAMMA)*5.*CPIN(I)*FMS**2)
POWER=1.+GAMMA)/GAMMA
AME1V(20)=((PIV(I)/ PT)**POWER=1.)**2.**(GAMMA=1.)
AME2V(I)=SQRT(AME1V(I))
30 UEV(20)=AME1V(I)*AT/SQRT(1.**(5.**(GAMMA=1.)**(AME1V(I)=2))
40 DELP8I=0.05
C ASSUME AN INITIAL SEPARATION ANGLE, PSIOLOD AND AN
C INCREMENT, DELP8I

IL=IL+1
IUE=IUE+1
P8=P8V(IU)
PB=P8*(1.+GAMMA)*5.*CPIN(IL+1)*FMS**2)
POWER=1.+GAMMA)/GAMMA
AME1=AME1V(IL)
AME2=((P2/PT)**POWER=1.)**2.**(GAMMA=1.)
AME3=SQRT(AME2)
UE=UEV(IL)
SIGMA=12.*((1.+2298*AME1)
I8D=0
I=0
PSIOLO=0
IF (IL.GT.1) P8IOLOD=P8IV(IL=1)=DELP8I
P8IOLOD=PSIOLO+DELP8I
50 CONTINUE
I=I+1
C CALCULATE M1

C IF (IL.GT.1) OR (I.GT.1) GO TO 60
XSTR(IL)=XL(IL)
APPENDIX

Y8TR(IL) = RAD(IL)

60 CONTINUE

IF (IL=2) TO 80, 90

70 M1=0,

GO TO 90

80 ANGLE1 = P5IV1(1)

M1TAN(ANGLE1) = SQR((XL(2) - XL(1))**2 + (RAD(2) - RAD(1))**2)

90 CONTINUE

RHO0=1.

RHO1=1.

RHO2=1.

IF (I .GE. 100) GO TO 260

UBUE=0.

ALP1 = C2090 + 0.226 * AME1 + 0.08 * UBUE / SIGMA

ALPHA = SIGMA / ALP1

DELTA = SIGMA / FLOA(NPT=1)

ICNT=0

100 ICNT=ICNT+1

UBUED=UBUE

SUM1=0.

SUM2=0.

INTJ=INTJ-1

IF (ICNT.EQ.1) GO TO 110

ANUM = 1.0 * (GAMMA-1.0) * 0.5 * ((1.0 - UBUE) * XERFO - UBUE)**2 / (UE**AT)**2

POW = 1.0 / (GAMMA-1.0)

DEN = 1.0 * (GAMMA-1.0) * 0.5 * (UE**AT)**2

RH10 = ANUM / DEN

RH11 = ANUM / DEN

RH12 = ANUM / DEN

110 CONTINUE

AX1 = 1.0 * M1 * THET2 / (RAD(IL) * PSIO1D)

AX2 = 1.0 * M1 * THET2 / (RAD(IL) * PSIO1D)

SUM1 = SUM1 + (DELTA / 3.0) * (RH00 * XERFO * AX0 + 4.0 * RHO1 * XERF1 * AX1 + RHO2 * XERF2 * AX2)

1AX2)

SUM2 = SUM2 + (DELTA / 3.0) * (RH00 * XERFO * AX0 + 4.0 * RHO1 * XERF1 * AX1 + RHO2 * XERF2 * AX2)

1AX2)

120 CONTINUE

UBUE=SUM1 / SUM2

IF (ICNT .GT. 10) GO TO 130

IF (ABS(UBUE-UBUE) .GT. 0.001 * (UBUEO-UBUE) * 5) GO TO 100

C

THETA ITERATION

C

130 THETA=0.

IF (UBUE .GT. 1.0) UBUE=1.0

ICNT=0

DELTH = DELTA

RIGHT = UBUE / (1.0 - UBUE) = 1.

140 ARG = SIGMA * (THETA - ALPHA)


APPENDIX

\begin{verbatim}
ALEFT*ERT(ARG) IF (ICNT,GE,100) GO TO 170 IF (AB(AEFT),GT,AB(RIGHT)) GO TO 150 IF (AB(RIGHT)=ALEFT),LE,AB(01)*(RIGHT-ALEFT)*.5)) GO TO 160 THETA+THETA=DELTH DELTH=DELTH/10, ICNT=ICNT+1 IF (THETA,GE,PSIOLD) GO TO 160 GO TO 140 CONTINUE ICNT=THETA/PSIOLD*100 IF (ICNT/2,HE,ICNT) ICNT=ICNT+1 CONTINUE SUM3=0, USE SIMPSON8 RULE TO INTEGRATE THE MOMENTUM EQUATION DO 180 J=2,ICNT,2 THETO=Delta*(J-2) THETO=DELTA*(J=1) THETO=DELTA*(J=0) ARG0=SIGMA*(THETO=ALPHA) ARG1=SIGMA*(THETO1=ALPHA) ARG2=SIGMA*(THETO2=ALPHA) XERFO,5=(1.+ERT(ARG0)) XERF1,5=(1.+ERT(ARG1)) XERF2,5=(1.+ERT(ARG2)) ANUM1,=(GAMMA=1.),*,5*,((1.+UBUE)*XERFO-UBUE)**2*(UE/AT)**2 POWER1,1,(GAMMA=1.,) DEN1,1=(GAMMA=1.),5*(UE/AT)**2 RHO=ANUM/DEN**POWER ANUM1,=(GAMMA=1.,),5*,((1.+UBUE)*XERF1-UBUE)**2*(UE/AT)**2 RHO1=ANUM/DEN**POWER ANUM1,=(GAMMA=1.,),5*,((1.+UBUE)*XERF2-UBUE)**2*(UE/AT)**2 RHO2=ANUM/DEN**POWER AXO1,=(H1*THETO)/(RAD(IL)*PSIOLD) AX1,1=(H1*THETO1)/(RAD(IL)*PSIOLD) AX2,1=(H1*THETO2)/(RAD(IL)*PSIOLD) TEMP=DELTA/3.)*(RHO0+COS(THETO)*((1.+UBUE)*XERFO-UBUE)**2*AXO+4.*RHO2+COS(THETO2)*((1.+UBUE)*XERF2-UBUE)**2*AX1+RHO2*COS(THETO2)*((1.+UBUE)*XERF2-UBUE)**2*AX2) SUM3=SUM3+TEMP CONTINUE SUMM=0, IF (ICNT,GE,100) GO TO 200 ICNT=ICNT+2 DO 190 J=ICNT,INTJ,2 THETO=Delta*(J=2) THETO=DELTA*(J=1) THETO=DELTA*(J=0) ARG0=SIGMA*(THETO=ALPHA) ARG1=SIGMA*(THETO1=ALPHA) ARG2=SIGMA*(THETO2=ALPHA) XERFO,5=(1.+ERT(ARG0)) XERF1,5=(1.+ERT(ARG1)) XERF2,5=(1.+ERT(ARG2)) ANUM1,=(GAMMA=1.,),5*,((1.+UBUE)*XERFO-UBUE)**2*(UE/AT)**2 POWER1,1,(GAMMA=1.,) DEN1,1=(GAMMA=1.,) RHO=ANUM/DEN**POWER ANUM1,=(GAMMA=1.,),5*,((1.+UBUE)*XERF1-UBUE)**2*(UE/AT)**2 RHO1=ANUM/DEN**POWER ANUM1,=(GAMMA=1.,),5*,((1.+UBUE)*XERF2-UBUE)**2*(UE/AT)**2 RHO2=ANUM/DEN**POWER AXO1,=(H1*THETO)/(RAD(IL)*PSIOLD)
\end{verbatim}
APPENDIX

AX1 = (H1 * THET1) / (RAD(IL) * PSIOLD)
AX2 = (H1 * THET2) / (RAD(IL) * PSIOLD)

TEMP = (DELTA/3) * (RHOCOS(THET1) * (1. + UBE) * XERF0 / UBE) * 2 + AX1 * RHOCOS(THET2) * ((1. + UBE) / UBE) * 2 + AX1 * RHOCOS(THET2) * (1. + UBE) * XERF1 / UBE)

SUM = SUM + TEMP

190 CONTINUE

200 CONTINUE

SUM = SUM + SUM + SUM

ANUM1 = (GAMMA = 1) * 5 * (1 + UBE) * 2 * (UE / AT) * 2
DEN1 = (GAMMA = 1) * 5 * (UE / AT) * 2

RHT = 0.005821 * PT / TO

RNHRT = (1 + (GAMMA = 1) * 5 * AME1 / 2) * (1 / (1 + GAMMA))

RNHRT = RNHRT / RHT

ANUM2 = (GAMMA = 1) * 5 * UBE * 2 * (UE / AT) * 2

AMACB = AMACB / RNHRT / UBE * 2

CF = C1 = 0.2

GO TO 220

210 CONTINUE

ENUM = 56 / 3600.

XX1 = 1.

RXX = UBE / UBE / ENUM

CF = 1.32 / SQRT (RXX)

SKIN = CF * 5 * AMACB

DPDX = (P2 - P1) / (XL(IL + 1) - XL(IL))

H2 = H1

DIST = SQRT ((XL(IL + 1) - XL(IL)) * 2 + (RAD(IL + 1) - RAD(IL)) * 2)

IF (IL, EQ, 1) GO TO 230

H2 = H1 / TAN(PHI) / TAN(PBI(1) + OIST * TAN(PBI(1))

C AB PERPENDICULAR DISTANCE FROM SEPARATION SLOPE LINE TO CONTOUR

C DB DISTANCE ALONG SLOPE LINE AT SEPARATION

DB = ABS(H2 - H1) / TAN(PBI(1))

SLP = (RAD(IL) = RAD(IL + 1)) / (XL(IL + 1) - XL(IL))

SLP = TAN(SLOP)

AMUS = SLOP = ABODSV

AMU = TAN(AMUS)

H2 = H1

C COMPARE THE RIGHT AND LEFT SIDE OF MOMENTUM EQUATION

C

SUM = SUM + SUM + SUM

SUM = SUM + SUM + SUM

SUM = SUM + SUM + SUM

SUM = SUM + SUM + SUM

IF (IL, EQ, 1) GO TO 240

X01 = H1 / PBIV(1) = 1

X02 = H2 / PBIV

RATIO = (1 + H1 / RAD(IL)) / (1 + H1 / RAD(IL)) / (1 + H1 / RAD(IL - 1) / RAD(1))

SUM = (X02 * AMULT + SUM = SUM * X01 * RATIO * SUM) / (X02 + X01)

240 CONTINUE

GO TO 300
APPENDIX

250 CALL NEWRAP(I,PBIOLD,G,LI,EP8LN,PSINEW,IJB)

IF (PSINEW_L,T,0.) GO TO 260
IF (PSINEW_GT,AB0DSV) GO TO 260
IF (I,EQ,100) GO TO 260
GO TO 300

C
ITERATION FAILED, USE ANGLE FROM PREVIOUS ITERATION

C

260 CONTINUE
IF (AME1_L,T,2) GO TO 280
IF (AME1_L,T,5) GO TO 270
SLOPE=(8,6-15,625)/(1.,5)
PSINEW=15,625+SLOPEx(AME1-5)
GO TO 290

270 SLOPE=(15,625-17,5)/(1.,5)
PSINEW=17,5+SLOPEx(AME1-2)
GO TO 290

280 PSINEW=17,5
WRITE (6,370) PSINEW
P8INEM=P8INEM+DEGRAD
IF (P8INEW_GT,AB0DSV) P8INEW=AB0DSV
P8IOLD=P8INEW
BEGIN=P8INEW
IF (AMG.EQ.9.) WRITE (6,380)
GO TO 300
P8IOLD=P8INEW
IF (IJB.EQ.0) GO TO 50

C
CALCULATE DISCRIMINATING STREAMLINE

C

300 P8IOLD=P8INEW
IF (IJB,EQ,O) GO TO 50
DELOLD=DELLOC
IF (IL_GT,1) DELLOC=ABB(H2=H1)/DB
DELLOC=ATAN(DELLOC)
IF (IL_EQ,1) DELLOC=P8IOLD
IF (IL,EQ,1) DELOLD=0.
ASTR=ABOD-(DELLOC-DELOLD)
WRITE (6,370) P8INEW
P8INEM=P8INEM+DEGRAD
IF (P8INEW_GT,AB0DSV) P8INEW=AB0DSV
P8IOLD=P8INEW
IF (AMG.EQ.9.) WRITE (6,380)
GO TO 300

310 YSTR(JB+1)=RAD(JB)
CONTINUE

320 CONTINUE
SUM8V=SUM5
UBUEV(IL)=UBUE
H1V(IL)=H1
P8IV(IL)=P8IOLD
AB0D=ASTR
P8IV(P8IOLD)
IF (ISTOP.EQ.n) GO TO 160
IF (IJET.EQ.O) GO TO 310

56
IF (IL+1, LT, NEXT) GO TO 340

CALL JET ENTRAINMENT IF OPTION TURNED ON

RL=RAD(IL+1)
RD=STR(IL+1)
NST=NST+NEXT+1
DO 330 I=NEXT, NST
330 X=JET(I)*X(I)=X(NEXT)
UM=UBUV(IL)*UE
ORDDXX=YSR(NEXT)*YSR(NEXT-1)/(X(NEXT)*X(NEXT-1))
CALL JET (RL, UM, C, DR, DX, DUM, RX, N, B, X, JET(NEXT), UEV(NEXT), YSR(NEXT), R
1ST), RI, UJ, DR, DXX)
GO TO 360
340 CONTINUE

DETERMINE IF ITERATION COMPLETE

IF (IL+1, EQ, 1) GO TO 350
DB=AABS(MH=HI)/TAN(PSI/OLD)
AMU=SLPL=ABD/BV
AR=DB*TAN(AMU)
HI=MH+AR
IF (HI, GT, 0.) GO TO 350
IF (IL+1, EQ, 0) IL=IL
HI=HTY(IL)*.01
IF (IL+1, LT, NST) GO TO 40
360 RETURN

FORM 1 (1H, 6HPSINUEF12, 4)
370 FORM (1H, 6HITERATION FOR DISCRIMINATING STREAMLINE ANGLE, 2BFREE)
380 TIALED, USING DEFAULT VALUE, 17H TRY DECREASING, 36HSTEP SIZE (HOB)
2RE POINTS ON AFTERBODY)
390 END

FUNCTION ERT (X)
C
ERT
C THIS FUNCTION ROUTINE OBTAINS VALUE OF THE ERROR FUNCTION ERT 3
C WITH ARGUMENT X USING LIBRARY SUBROUTINE ERF
C
CALL ERF (X,Y)
ERT=Y
RETURN

END ERT

SUBROUTINE NEWRAP (ICNT, X, FUNC, TOLL, XZERO, IE)
C
IE=0
IF (ICNT, GT, 100) STOP
IF (ICNT=2) 10, 20, 30
10 FUNC=FUNC
X1=X
IF (X1, EQ, 0.) X1=100.*TOLL
XZERO=X1+.1*X
GO TO 90
20 CONTINUE
FUNC=FUNC
X2=X
GO TO 80
30 CONTINUE
IF (FUNC=FUNC) 50, 40, 40
40 FUNC=FUNC
X1=X2
X2=X
GO TO 80
50 CONTINUE
APPENDIX

IF (FUNC*FUNC) 70,80,60
60 FUNC=FUNC
X2=X
GO TO 60
70 FUNC=FUNC
X1=X
C CALCULATE DERIVATIVE
C DERM(FUN2=FUN1)/(X2=X1)
XZERO*X2=FUN2/DERV
IF (ABS(XZERO-X).LT.(ABS((XZERO*X2)*.5)*TOLL) IER=1
RETURN
END
SUBROUTINE JET (RJ,UM,C,D,DUJ,DUMOX,ROINS,T,XIN,U,U8TR,RJA,UJA,DRJET)
100X0) JET 2
10 CONTINUE JET 26
20 CONTINUE JET 29
25 CONTINUE JET 30
30 CONTINUE JET 31
31 CONTINUE JET 32
32 CONTINUE JET 33
33 CONTINUE JET 34
34 CONTINUE JET 35
35 CONTINUE JET 36
36 CONTINUE JET 37
37 CONTINUE JET 38
38 CONTINUE JET 39
39 CONTINUE JET 40
40 CONTINUE JET 41
41 CONTINUE JET 42
42 CONTINUE JET 43
43 CONTINUE JET 44
44 CONTINUE JET 45
45 CONTINUE JET 46
46 CONTINUE JET 47
47 CONTINUE JET 48
48 CONTINUE JET 49
49 CONTINUE JET 50
50 CONTINUE JET 51
51 CONTINUE JET 52
52 CONTINUE
APPENDIX

11, 5 eta ** 3)) * (d D O X = D L D X = E T A = D L D X ) + (R = (U M * E T A + E L U E * (2, 2, 5, 5 J ) E T A )

2 * (2, 2, 5, 5 * e t a + e l u e * (2, 2, 5, 5 * e t a + e l u e )

3 * (2, 2, 5, 5 * e t a + e l u e ) * D L D X )

T R M 2 * (U M * E T A + E L U E * (2, 2, 5, 5 * e t a + e l u e )

1 + (R = (U M * E T A + E L U E * (2, 2, 5, 5 * e t a + e l u e )

2 * (2, 2, 5, 5 * e t a + e l u e ) * D L D X )

B = T R M 1 + T R M 2

A = (U M * E T A + E L U E * (2, 2, 5, 5 * e t a + e l u e )

1 + (R = (U M * E T A + E L U E * (2, 2, 5, 5 * e t a + e l u e )

2 ** 2 * (e t a + e l u e ) * D L D X )

3 * (e t a + e l u e ) * D L D X )

b = T R M 3

C O R D D X + L I D X

D = T R M 1 + T R M 2 + T R M 3

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118
TRM2=1*UM**2+1*UM*DELUE+.4156*DELUE**2*DRDCX+L1**2*(UM+.7428*JET)
+DELUE*SUM DX
TRM3=1*RC+2*UM+1*DELUE*SUM DX+L1**2*(.6192*DELUE+.7428*UM)*DDU
DELUE+RC*SUM DX
TRM4=SUM DX+L1**2*CE+UM*DELUE*.5*U1*SUM DX
U2DELUE=TRM1+TRM2+TRM3
AMM=DELUX+JIT 1/EIL (U10*SUM DX+UM)
XMONT=U1*RA+RDX+(U2DRX+U2DRX)*RE+2.5*SUM DX+UE1*SUM DX=UE1*AMM
CALL NEWRAP (ICNT, DUMDX, XMONT, TOL, DUMDXN, IE)
IF (ICNT.LE.30) GO TO 60
1STA1=STA1+1
GO TO 130
60 CONTINUE
IF (IE, NE, 0) GO TO 70
DUMDX=DUMDXN
GO TO 20
70 CONTINUE
C
C SOLVE FOR THE RADIUS OF THE DISCRIMINATING STREAMLINE
C
X=XIN(ISTA)
DX=XIN(ISTA)-XIN(ISTA-1)
L1=L2
IF (ETA, LE, ETA2) GO TO 80
IF (DRDX, EQ, 0.) GO TO 80
RD=DRDX*DX+RD
ETA=ETA+DELUE*1/(SUM DX+UM)
ETA2=1/(1E10*SUM DX+UM)
IF (ETA, LT, ETA2) GO TO 80
C
80 CONTINUE
DRODX=0.
UM=DUMDX+UM
IF (UM, GE, 0.) GO TO 170
RJ=RD+RDX+L1
DRE=RD+RDX+L2
RE=RE+DRE+DUX
IF (DRDX, NE, 0.) GO TO 100
LI=LI=RA=L2
IF (L1, LE, 0.) GO TO 170
DELUE=1/(SUM DX+UM)
ETA1=1/(1E10*SUM DX+UM)
RAD=RJ+L2+ETA*L1
IF (RD, GT, RTSV) GO TO 170
100 CONTINUE
YSTR(ISTA)=RD+RJA(ISTA)=RO=L2
RD=RD+RJA(ISTA)
ICNT=0.
IF (RO, GE, RO) GO TO 110
IF (YSTR(ISTA), LE, RJA(ISTA)) GO TO 110
C
110 ICNT=ICNT+1
DO 120 ICNT, NIST
120 YSTR(I)=RJA(I)
C
GO TO 200
130 SLOPE=(YSTR(ISTA)=1*YSTR(ISTA-2))/DX
IF (SLOPE, GT, 0.) GO TO 150
130 YSTR(I)=YSTR(I)
DO 140 ICNT, NIST
140 YSTR(I)=YSTR(I-1)
C
GO TO 200
150 DO 160 ICNT, NIST
160 YSTR(I)=YSTR(I-1)
C
GO TO 200
170 DO 180 ICNT, NIST
180 GO TO 200
C
C
L2=C*XIN(I)
YSTR(I)=RD8V+RJA(I)-RO=L2
180 IF (YSTR(I)<LT.RJA(I)) YSTR(I)=RJA(I)
GO TO 200
190 CONTINUE
IF (ISTA.LT.N8IN) 00 TO 10
JET 187
200 RETURN
JET 190
END
SUBROUTINE SMINT (XA,YA,NA,NMIN,NMAX)
COMMON /SAVE/ 3B(201),SC(201),YJBC(201),XIN
DIMENSION X1(201), X2(201), Y1(201), Y2(201), XA(1), YA(1), B(201)
1. 61(201), Y22(201), Y23(201), Y(201), Z(201), DDY(201), DY(201)
2. DZ(201), DDZ(201), Z(201)
C
NA1=NA1-1
DO 10 I=1,NA1
X1(I)=XA(I)
X2(I)=X(A(I+1))
Y1(I)=YA(I)
Y2(I)=Y(A(I+1))
K9=1
NSMTH11=NMIN+1
IF (NSMTH11.LT.2) NSMTH11=2
NSMTH21=NA1-1
DO 20 I=NA1,NA
IF (XIN.UX.XA(I)) GO TO 30
NSMTH21=I+6
20 CONTINUE
DO 30 IF (NSMTH21.GT.NA-1) NSMTH21=NA1
30 IF (N8MTH2.GT.NA-1) N8MTH2=NA1
RETURN
END
SUBROUTINE SMOOTH (X1,X2,Y1,Y2,K9,K11,N8MTH1,N8MTH2,IV8M,NA,B,1,Y22,Y23)
CALL SMOOTH (X1,X2,Y1,Y2,K9,K11,N8MTH1,N8MTH2,IV8M,NA,B,1,Y22,Y23)
DO 40 I=1,NA1
X1(I)=X1(I)
YA(I)=YA(I)
RETURN
END
SUBROUTINE SMOOTH (X1,X2,Y1,Y2,K9,K11,N8MTH1,N8MTH2,IV8M,NA,B,1,Y22,Y23)
DIMENSION X1(NA), Y1(NA), Y2(NA), B(NA), B1(NA), X2(NA), Y22(NA), Y23(NA)
1. Y23(NA), Y(NA), Z(NA), NSMTH1(5), NSMTH2(5), DDY(NA), DY(NA), DZ(NA)
2. DZ(NA), Z(NA)
C
DO 170 I=1,K9
N5=NSMTH1(I)
N6=NSMTH2(I)
I=MIN0(N5,N6)
T2=MAX0(N5,N6)
B(I)=0.0
TF (IV8M.EQ.1) GO TO 20
DO 10 J=I1,12
10 B(J)=B(J-1)+8*RT((X2(J)-X(J))**2+(Y2(J)-Y(J))**2)
GO TO 20
DO 30 J=I1,12
30 B(J)=B(J-1)+8*RT((X1(J)-X(J-1))**2+(X2(J)-X(J-1))**2)
CONTINUE
I=I2+I1
DELS=5(I2)/II
T3=I2+1
S1(I1)=0.0
DO 50 J=I1,I3
50 CONTINUE
APPENDIX

50  B1(J)=B1(J-1)+DEL8
B1(I2)=B1(I2)
DO 90 J=I1,13
NO A0 K=I1,13
IF (B(K)=B1(J)) 60,70,70
60  CONTINUE
70  IF (IVSM.EQ.1) GO TO 80
Y22(J)=Y1(K)+Y2(K)+Y1(K))*(B1(J)=B(K)=B1(K))/B(K)*B(K-1)
GO TO 90
80  Y22(J)=Y1(K)+Y2(K)+Y1(K))*(B1(J)=B(K)=B1(K))/B(K)*B(K-1)
90  CONTINUE
Y22(I2)=Y2(I2)
IF (IVSM.EQ.1) Y22(I2)=Y1(I2)
100  CALL R8MTH (Y,Z,N,K,NA,DDY,DDZ)
DO 110 J=I1,12
Y22(J)=Z(J-1)=Z(J-1)
IF (IVSM.EQ.1) Y22(J)=Y1(J)
DO 160 J=I1,13
DO 120 K=I1,IS
IF (B1(K)=B1(J)) 120,150,150
120  CONTINUE
150  IF (IVSM.EQ.1) GO TO 140
Y2(J)=Y28(K)-1)+Y28(K)=Y28(K-1))*(B1(J)=B(K)=B1(K))/B(K)*B(K-1)
GO TO 150
140  CONTINUE
Y1(J)=Y28(K-1)+Y28(K)=Y28(K-1))*(B1(J)=B(K)=B1(K))/B(K)*B(K-1)
150  CONTINUE
IF (IVSM.EQ.0) Y1(J)=Y2(J)
160  CONTINUE
170  CONTINUE
RETURN
180  END

SUBROUTINE R8MTH (Y,Z,N,K,NA,DDY,DDZ)
DIMENSION Y(NA), Z(NA)
IF (N<5) 40,10,10
10  CALL R8MTH (Y,N,K,NA,Z)
NM1=N-1
NM2=N-2
DO 20 I=1,NM1
DY(I)=Y(I+1)-Y(I)
20  CONTINUE
DO 30 I=1,NM2
DDY(I)=DY(I+1)-DY(I)
30  CONTINUE
RETURN
30  END

SUBROUTINE R8MTH (Y,N,K,NA,Z)
DIMENSION Y(NA), Z(NA)
J=0
DO 10 I=1,1
Z(I)=Y(I)
10  CONTINUE
Z(I)=Y(I)
20  J=J+1
CALL R8MTH (Z,NA,U,Z,NA)
IF (U) 60,60,30
APPENDIX

SUBROUTINE STHM(Y, N, U, Z, NA)

DIMENSION Y(NA), Z(NA)

F(ETA) = (1.0 - ETA * ETA)**2
G(ETA) = 0.5 * ETA

U = 0.0
J = 3

IF (A - ABS(D)) 20, 40, «0
F = F(A/D)
A = A * ABS(Y(J - 1) + Y(J + 1) - Y(J))
B = ABS(3.0 * CY(J - 1) + 2.0 * Y(J))
TF(A, B) 30, «0, «0
GS = G(A/ABS(D))
Z(2) = Y(2) - 0.5 * FS * GS * D
GO TO 50

J = J + 1
IF (J = (N - 2)) 10, 10, 60
A = 0.4 * A5B(Y(4) - Z(4))
B = 2.0 * Y(1) + 10.0 * Y(2) - 2.0 * Y(3) + Y(4) - Z(4)
TF(A = A5B(D)) 70, 80, 80
F = F(A/D)
G = G(A/ABS(D))
Z(2) = Y(2) - 0.5 * FS * GS * D
GO TO 50

Z(2) = Y(2)

A5B(Y(N - 3) = Z(N - 3))
D = 2.0 * Y(N - 3) - Z(N - 3) - 2.0 * Y(N - 2) + 10.0 * Y(N - 1)
TF(A = A5B(D)) 100, 110, 110
F = F(A/D)
G = G(A/ABS(D))
Z(N - 1) = Y(N - 1) - 0.5 * FS * GS * D
GO TO 120

Z(N - 1) = Y(N - 1)
120 RETURN
END

SUBROUTINE IUNI(NMAX, N, X, NTAB, Y, IORDER, X0, Y0, IPT, IERR)

C*
C* SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C* LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C* OF A SET OF FUNCTIONS AT A POINT X0. IUNI
C* USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C* VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C* THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C* AT EQUAL OR UNEQUAL INTERVALS, EACH DEPENDENT
C* VARIOUS TABLE MUST CONTAIN FUNCTION VALUES CORRESPONDING
C* TO EACH X(J) IN THE INDEPENDENT VARIABLE TABLE.
C* THE ESTIMATED VALUES ARE RETURNED IN THE YO ARRAY WITH
C* THE N-TH VALUE OF THE ARRAY HOLDING THE
C* VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C**************************************************************************************************
C* PURPOSE:
C* SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C* LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C* OF A SET OF FUNCTIONS AT A POINT X0. IUNI
C* USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C* VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C* THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C* AT EQUAL OR UNEQUAL INTERVALS, EACH DEPENDENT
C* VARIOUS TABLE MUST CONTAIN FUNCTION VALUES CORRESPONDING
C* TO EACH X(J) IN THE INDEPENDENT VARIABLE TABLE.
C* THE ESTIMATED VALUES ARE RETURNED IN THE YO ARRAY WITH
C* THE N-TH VALUE OF THE ARRAY HOLDING THE
C* VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C**************************************************************************************************
C* USE:
C* SUBROUTINE IUNI(NMAX, N, X, NTAB, Y, IORDER, X0, Y0, IPT, IERR)
C**************************************************************************************************

C**************************************************************************************************
C* PURPOSE:
C* SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C* LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C* OF A SET OF FUNCTIONS AT A POINT X0. IUNI
C* USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C* VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C* THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C* AT EQUAL OR UNEQUAL INTERVALS, EACH DEPENDENT
C* VARIOUS TABLE MUST CONTAIN FUNCTION VALUES CORRESPONDING
C* TO EACH X(J) IN THE INDEPENDENT VARIABLE TABLE.
C* THE ESTIMATED VALUES ARE RETURNED IN THE YO ARRAY WITH
C* THE N-TH VALUE OF THE ARRAY HOLDING THE
C* VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C**************************************************************************************************
C* USE:
C* SUBROUTINE IUNI(NMAX, N, X, NTAB, Y, IORDER, X0, Y0, IPT, IERR)
C**************************************************************************************************

C**************************************************************************************************
C* PURPOSE:
C* SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C* LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C* OF A SET OF FUNCTIONS AT A POINT X0. IUNI
C* USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C* VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C* THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C* AT EQUAL OR UNEQUAL INTERVALS, EACH DEPENDENT
C* VARIOUS TABLE MUST CONTAIN FUNCTION VALUES CORRESPONDING
C* TO EACH X(J) IN THE INDEPENDENT VARIABLE TABLE.
C* THE ESTIMATED VALUES ARE RETURNED IN THE YO ARRAY WITH
C* THE N-TH VALUE OF THE ARRAY HOLDING THE
C* VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C**************************************************************************************************
C* USE:
C* SUBROUTINE IUNI(NMAX, N, X, NTAB, Y, IORDER, X0, Y0, IPT, IERR)
C*************************************************************************************************

C**************************************************************************************************
C* PURPOSE:
C* SUBROUTINE IUNI USES FIRST OR SECOND ORDER
C* LAGRANGIAN INTERPOLATION TO ESTIMATE THE VALUES
C* OF A SET OF FUNCTIONS AT A POINT X0. IUNI
C* USES ONE INDEPENDENT VARIABLE TABLE AND A DEPENDENT
C* VARIABLE TABLE FOR EACH FUNCTION TO BE EVALUATED.
C* THE ROUTINE ACCEPTS THE INDEPENDENT VARIABLES SPACED
C* AT EQUAL OR UNEQUAL INTERVALS, EACH DEPENDENT
C* VARIOUS TABLE MUST CONTAIN FUNCTION VALUES CORRESPONDING
C* TO EACH X(J) IN THE INDEPENDENT VARIABLE TABLE.
C* THE ESTIMATED VALUES ARE RETURNED IN THE YO ARRAY WITH
C* THE N-TH VALUE OF THE ARRAY HOLDING THE
C* VALUE OF THE N-TH FUNCTION VALUE EVALUATED AT X0.
C*
C**************************************************************************************************
C* USE:
C* SUBROUTINE IUNI(NMAX, N, X, NTAB, Y, IORDER, X0, Y0, IPT, IERR)
APPENDIX

CALL IUNI(NMAX,N,X,NTAB,Y,IORDER,XO,YO,IPT,IERR)

PARAMETERS:

NMAX The maximum number of points in the independent variable array.

N The actual number of points in the independent array, where N .LE. NMAX.

X A one-dimensional array, dimensioned (NMAX) in the calling program, which contains the independent variables. These values must be strictly monotonic.

NTAB The number of dependent variable tables.

Y A two-dimensional array, dimensioned (NMAX,NTAB) in the calling program, each column of the array contains a dependent variable table.

IORDER Interpolation parameter supplied by the user.

#0 Zero order interpolation; the first function value in each dependent variable table is assigned to the corresponding member of the YO array. The functional value is estimated to remain constant and equal to the nearest known function value.

XO The input point at which interpolation will be performed.

YO A one-dimensional array, dimensioned (NTAB) in the calling program. Upon return the array contains the estimated value of each function at xo.

IPT On the first call IPT must be initialized to -1 so that monotonicity will be checked. Upon leaving the routine IPT equals the value of the index of the X value preceding XO unless extrapolation was performed. In that case the value of IPT is returned as:

#0 Denotes XO .LT. X(J) if the X array is in increasing order and X(J) .GT. XO if the X array is in decreasing order.

#N Denotes XO .GT. X(N) if the X array is in increasing order and XO .LT. X(N) if the X array is in decreasing order.

On subsequent calls, IPT is used as a pointer to begin the search for xo.

IERR Error parameter generated by the routine:

#0 Normal return.

#J The J-th element of the X array is out of order.

#1 Zero order interpolation performed because only one point was in X array.

#2 Zero order interpolation performed because only one point was in X array.

#3 No interpolation was performed because insufficient points were supplied for second order interpolation.

#4 Extrapolation was performed.

Upon return the parameter IERR should be tested in the calling program.
TEST FOR ZERO ORDER INTERPOLATION

DELX=X(2)-X(1)
IF (ORDER EQ 0) GO TO 10
IF (N LT 2) GO TO 20
GO TO 50

10 TERR=1
GO TO 30

20 TERR=2

DO 40 NTM1,NTAB
YO(NT)=Y(1,NT)
40 CONTINUE
RETURN

IF (IPT .GT. 0) GO TO 65
CHECK FOR TABLE OF NODE POINTS BEING STRICTLY MONOTONIC
THE SIGN OF DELX SIGNIFIES WHETHER TABLE IS IN
INCREASING OR DECREASING ORDER.

IF (DELX .EQ. 0) GO TO 190
IF (N .EQ. 2) GO TO 65
CHECK FOR SIGN CONSISTENCY IN THE DIFFERENCES OF
SUBSEQUENT PAIRS

DO 60 J=2,NM1
IF (DELX .LT. 0) X(J+1)=X(J)+DELX
CONTINUE

IPT IS INITIALIZED TO BE WITHIN THE INTERVAL

65 IF (IPT .LT. 1) IPT=1
IF (IPT .GT. NM1) IPT=N1
IN=SIGN ((1,0),DELX)*X(J=IPT))
70 P=X(IPT)=X(IPT+1)=X(J)
80 IPT=IPT+1

TEST TO SEE IF IT IS NECESSARY TO EXTRAPOLATE

IF (IPT .GT. 0 .AND. IPT .LT. N) GO TO 70
TERR=4
IPT=IPT+1

TEST FOR ORDER OF INTERPOLATION
APPENDIX

90 IF (IORDER .GT. 1) GO TO 120

FIRST ORDER INTERPOLATION

IPT1=IPT+1
XTMP1=x[IPT]
XTMP2=x[IPT+1]
XTMP3=x[IPT+2]
DO 100 NT=1,NTAB
YTMP=V(IPT,NT)-V(IPT,NT)+YTMP*XTMP1
100 CONTINUE
IF (IERR .EQ. 0) IPT=IPT+1
RETURN

SECOND ORDER INTERPOLATION

120 IF (N .EQ. 0) GO TO 200

CHOOSING A THIRD POINT SO AS TO MINIMIZE THE DISTANCE BETWEEN THE THREE POINTS USED TO INTERPOLATE

IF (IPT .EQ. NM1) GO TO 130
IF (IPT .EQ. 1) GO TO 130
A1=A1+B(XO-X(IPT+1))
A2=A2+B(X(IPT+2)-XO)
IF (((A1-A2) .LT. 0), 130, 130
CONTINUE
130 L=IPT
140 L=IPT+1
150 V1=X(L)*X0
V2=X(L+1)*X0
V3=X(L+2)*X0
DO 160 NT=1,NTAB
YY1=(Y(L,NT)-V2-Y(L+1,NT)+V1)/(X(L+1)-X(L))
YY2=(Y(L+1,NT)-V1-Y(L+2,NT)+V2)/(X(L+2)-X(L+1))
YO(NT)=YY1*V3-YY2*V1)/(X(L+2)-X(L))
160 CONTINUE
IF (IERR .EQ. 0) IPT=IPT+1
RETURN

IERR IS SET TO THE SUBSCRIPT WHICH IS OUT OF ORDER

190 IERR=M+1
RETURN

200 IERR=3
RETURN

END
REFERENCES


Figure 1.- Analytical model of flow over nozzle boattail.
Figure 2. Comparison of one-dimensional jet exhaust flow calculation with method of Salas (ref. 14).
Figure 3.- Comparison of predicted boundary-layer characteristics with experiment of Winter, Rotta, and Smith (ref. 17). $M_\infty = 0.6$ and Reynolds number based on body length of $9.85 \times 10^6$. 

(a) Pressure distribution and body geometry.
(b) Displacement thickness, momentum thickness, and shape factor.

Figure 3.—Continued.

These are results from a stand-alone boundary-layer procedure (also next page).
(c) Skin friction.

Figure 3.- Concluded.
Figure 4.- Flow diagram of interaction procedure.
Figure 5.- Comparison of theory and experiment for flow over unseparated $1/D = 1.768$, $d_b/D = 0.51$ circular-arc nozzle with solid jet plume simulator. (Experimental data from ref. 22.)
(a) Separation location.

Figure 6.- Effect of separation location criteria. (Experimental data from ref. 21.)
(b) Predicted pressure distributions on $l/D = 0.8$, $d_b/D = 0.51$
  circular-arc boattail at $M_\infty = 0.8$.

Figure 6. Concluded.
Figure 7.- Effect of Mach number on flow over $l/D = 0.8$, $d_b/D = 0.51$ circular-arc boattail with solid jet plume simulator.
(b) Pressure drag coefficient.

Figure 7.- Concluded.
Figure 8.- Effect of afterbody closure on flow over $l/D = 1.0$ circular-arc boattails with solid jet plume simulators. $M_{\infty} = 0.8$. (Experimental data from ref. 23.)
(a) Pressure distributions for $l/D = 0.961$, $\delta_b/D = 0.51$
circular-arc conic boattail.

Figure 9.- Effect of Reynolds number on pressures and drag of afterbodies with solid jet plume simulators. $M_\infty = 0.6$. (Experimental data from ref. 1.)
(b) Pressure distributions for \( l/D = 0.95, \ d_b/D = 0.544 \) contoured boattail.

Figure 9.- Continued.
(c) Drag coefficient.

Figure 9.—Concluded.
Figure 10.— Effect of NPR on pressures and drag for $l/D = 0.8$, $d_b/D = 0.51$ circular-arc nozzle.

(a) Pressure distribution at $M_\infty = 0.6$. 

NPR Ref.
$\circ$ 2.0 24
$\square$ 2.9 22
$\diamond$ 5.0 22

Present method
(b) Pressure distribution at $M_\infty = 0.8$.

Figure 10.– Continued.
(c) Drag coefficient.

Figure 10.— Concluded.
Figure 11.- Comparison of experimental and predicted pressures for equivalent bodies of Berrier (ref. 25). $M_\infty = 0.8$ and NPR = 2.5.
Figure 12.- Flow chart for program DONBOL.
<table>
<thead>
<tr>
<th><strong>TEST CASE</strong></th>
<th>L/D=9</th>
<th>FOREBODY L/D=0.8</th>
<th>DB/D=6.5</th>
<th>CIRCULAR ARC NOZZLE</th>
<th>NPR=5%</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>1.07200</td>
<td>1.3244</td>
<td>0.1526</td>
<td>0.0182</td>
<td>5</td>
</tr>
<tr>
<td>1,000</td>
<td>33232.7</td>
<td>295.56</td>
<td>0.250</td>
<td></td>
<td>6.0000</td>
<td>8.0000</td>
</tr>
</tbody>
</table>

Figure 13.- Sample input data
DONBOL *** AN AXISYMMETRIC INVISCOID/VISCOID INTERACTION PROGRAM

BY LAWRENCE E. PUTNAM, NASA, LANGLEY RESEARCH CENTER

CASE TITLE = **TEST CASE** L/D=9 FOREBODY L/D=0.6 DB/D=0.91 CIRCULAR ARC NOZZLE NPR=5.03

***** CASE CONTROL DATA *****

OFF-BODY POINTS
LABUJERE COMPRESSIBILITY CORRECTION
MODIFIED REHOTKO TUCKER BOUNDARY LAYER SOLUTION
PRESS MODIFIED CONTROL VOLUME DISCRIMINATING STREAMLINE SOLUTION
PRESS CONTROL VOLUME SEPARATION LOCATION CRITERIA
START SEARCH FOR SEPARATION AT I = 113
END SEARCH FOR SEPARATION AT I = 135
JET EXHAUST PLUME CALCULATION
NOZZLE EXIT AT I = 135
SMOOTH AERODYNAMIC CONTOUR
SMOOTH PRESSURE DISTRIBUTION

FREE STREAM CONDITIONS
MACH NUMBER = 6.00
TOTAL PRESSURE = 100720,000 PASCALS
TOTAL TEMPERATURE = 324,440 KELVIN
REYNOLDS NUMBER = 12,182 MILLION PER METER

JET EXHAUST CONDITIONS AT NOZZLE EXIT
MACH NUMBER = 1.000
TOTAL PRESSURE = 332342,700 PASCALS
TOTAL TEMPERATURE = 295,560 KELVIN
NPR = 5,030

(a) Page 1.

Figure 14.— Sample output data.
FOR ITER = 0 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 1 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 2 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 3 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 4 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 5 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 6 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 7 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 8 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 9 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 10 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 11 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 12 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 13 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 14 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2
FOR ITER = 15 8 IERATIONS REQUIRED FOR CONVERGENCE IN MISNA2

(b) Page 2.

Figure 14.—Continued.
**TEST CASE** L/D = 9 FOREBODY L/D = 0.8 DB/D = 0.51 CIRCULAR ARC NOZZLE NPR = 0.03 ITERATION NO 15

MO = 5.000 TT = 324.64 KELVIN PT = 100720.0 PASCALS L = 152400 METERS SREP = 0.018242 SQ METERS

**BOUNDARY LAYER SEPARATION AT X/L = 9.529286** BOUNDARY LAYER REATTACHMENT AT X/L = 11.635000

<table>
<thead>
<tr>
<th>X/L</th>
<th>R/L</th>
<th>CP</th>
<th>CF</th>
<th>CDP</th>
<th>CDF</th>
<th>CDT</th>
<th>RDB/L</th>
<th>RCL</th>
<th>DEL/L</th>
<th>DEL/L</th>
<th>THETA/L</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>1.1704</td>
<td>0.0095</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0763</td>
<td>0.0190</td>
<td>0.4430</td>
<td>0.0095</td>
<td>0.0006</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2.0526</td>
<td>0.0380</td>
<td>0.2840</td>
<td>0.0067</td>
<td>0.0021</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.3777</td>
</tr>
<tr>
<td>2.0526</td>
<td>0.0380</td>
<td>0.2840</td>
<td>0.0067</td>
<td>0.0021</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.3777</td>
</tr>
</tbody>
</table>

Figure 14.--Continued.
**TEST CASE**  L/D=9  FOREBODY L/D=0.6  DB/D=0.51  CIRCULAR ARC NOZZLE  NPR=5.03  ITERATION NO 15

<table>
<thead>
<tr>
<th>X/L</th>
<th>R/L</th>
<th>CP</th>
<th>CF</th>
<th>CDP</th>
<th>CDF</th>
<th>CDT</th>
<th>RDS/L</th>
<th>RC/L</th>
<th>DF1/L</th>
<th>DEL/L</th>
<th>THETA/L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3390</td>
<td>5.000</td>
<td>.0133</td>
<td>.0024</td>
<td>.0022</td>
<td>.0529</td>
<td>.0507</td>
<td>.0000</td>
<td>.0107</td>
<td>.0689</td>
<td>.0067</td>
<td>1.5652</td>
<td></td>
</tr>
<tr>
<td>5.4133</td>
<td>5.000</td>
<td>.0131</td>
<td>.0024</td>
<td>.0022</td>
<td>.0536</td>
<td>.0514</td>
<td>.0000</td>
<td>.0108</td>
<td>.0700</td>
<td>.0068</td>
<td>1.5666</td>
<td></td>
</tr>
<tr>
<td>5.4915</td>
<td>5.000</td>
<td>.0130</td>
<td>.0024</td>
<td>.0022</td>
<td>.0543</td>
<td>.0522</td>
<td>.0000</td>
<td>.0110</td>
<td>.0710</td>
<td>.0069</td>
<td>1.5681</td>
<td></td>
</tr>
<tr>
<td>5.5678</td>
<td>5.000</td>
<td>.0126</td>
<td>.0024</td>
<td>.0022</td>
<td>.0551</td>
<td>.0530</td>
<td>.0000</td>
<td>.0111</td>
<td>.0720</td>
<td>.0070</td>
<td>1.5695</td>
<td></td>
</tr>
<tr>
<td>5.6441</td>
<td>5.000</td>
<td>.0127</td>
<td>.0024</td>
<td>.0022</td>
<td>.0558</td>
<td>.0536</td>
<td>.0000</td>
<td>.0113</td>
<td>.0730</td>
<td>.0071</td>
<td>1.5709</td>
<td></td>
</tr>
<tr>
<td>5.7203</td>
<td>5.000</td>
<td>.0127</td>
<td>.0024</td>
<td>.0022</td>
<td>.0565</td>
<td>.0544</td>
<td>.0000</td>
<td>.0114</td>
<td>.0740</td>
<td>.0072</td>
<td>1.5725</td>
<td></td>
</tr>
<tr>
<td>5.7966</td>
<td>5.000</td>
<td>.0126</td>
<td>.0024</td>
<td>.0022</td>
<td>.0573</td>
<td>.0551</td>
<td>.0000</td>
<td>.0116</td>
<td>.0750</td>
<td>.0073</td>
<td>1.5740</td>
<td></td>
</tr>
<tr>
<td>5.8729</td>
<td>5.000</td>
<td>.0126</td>
<td>.0024</td>
<td>.0022</td>
<td>.0580</td>
<td>.0558</td>
<td>.0000</td>
<td>.0117</td>
<td>.0760</td>
<td>.0074</td>
<td>1.5755</td>
<td></td>
</tr>
<tr>
<td>5.9492</td>
<td>5.000</td>
<td>.0126</td>
<td>.0024</td>
<td>.0022</td>
<td>.0587</td>
<td>.0566</td>
<td>.0000</td>
<td>.0118</td>
<td>.0770</td>
<td>.0075</td>
<td>1.5770</td>
<td></td>
</tr>
<tr>
<td>6.0254</td>
<td>5.000</td>
<td>.0127</td>
<td>.0024</td>
<td>.0022</td>
<td>.0594</td>
<td>.0573</td>
<td>.0000</td>
<td>.0119</td>
<td>.0780</td>
<td>.0076</td>
<td>1.5785</td>
<td></td>
</tr>
<tr>
<td>6.1017</td>
<td>5.000</td>
<td>.0127</td>
<td>.0024</td>
<td>.0022</td>
<td>.0602</td>
<td>.0580</td>
<td>.0000</td>
<td>.0120</td>
<td>.0790</td>
<td>.0077</td>
<td>1.5799</td>
<td></td>
</tr>
<tr>
<td>6.1780</td>
<td>5.000</td>
<td>.0128</td>
<td>.0024</td>
<td>.0022</td>
<td>.0609</td>
<td>.0587</td>
<td>.0000</td>
<td>.0122</td>
<td>.0800</td>
<td>.0078</td>
<td>1.5813</td>
<td></td>
</tr>
<tr>
<td>6.2542</td>
<td>5.000</td>
<td>.0130</td>
<td>.0024</td>
<td>.0022</td>
<td>.0616</td>
<td>.0594</td>
<td>.0000</td>
<td>.0124</td>
<td>.0810</td>
<td>.0079</td>
<td>1.5827</td>
<td></td>
</tr>
<tr>
<td>6.3305</td>
<td>5.000</td>
<td>.0131</td>
<td>.0024</td>
<td>.0022</td>
<td>.0623</td>
<td>.0602</td>
<td>.0000</td>
<td>.0125</td>
<td>.0820</td>
<td>.0080</td>
<td>1.5841</td>
<td></td>
</tr>
<tr>
<td>6.4068</td>
<td>5.000</td>
<td>.0133</td>
<td>.0024</td>
<td>.0022</td>
<td>.0630</td>
<td>.0609</td>
<td>.0000</td>
<td>.0126</td>
<td>.0827</td>
<td>.0081</td>
<td>1.5855</td>
<td></td>
</tr>
<tr>
<td>6.4831</td>
<td>5.000</td>
<td>.0135</td>
<td>.0024</td>
<td>.0022</td>
<td>.0637</td>
<td>.0616</td>
<td>.0000</td>
<td>.0128</td>
<td>.0837</td>
<td>.0082</td>
<td>1.5869</td>
<td></td>
</tr>
<tr>
<td>6.5593</td>
<td>5.000</td>
<td>.0136</td>
<td>.0024</td>
<td>.0022</td>
<td>.0644</td>
<td>.0623</td>
<td>.0000</td>
<td>.0129</td>
<td>.0846</td>
<td>.0083</td>
<td>1.5883</td>
<td></td>
</tr>
<tr>
<td>6.6356</td>
<td>5.000</td>
<td>.0137</td>
<td>.0024</td>
<td>.0022</td>
<td>.0651</td>
<td>.0630</td>
<td>.0000</td>
<td>.0130</td>
<td>.0856</td>
<td>.0084</td>
<td>1.5897</td>
<td></td>
</tr>
<tr>
<td>6.7119</td>
<td>5.000</td>
<td>.0144</td>
<td>.0024</td>
<td>.0022</td>
<td>.0658</td>
<td>.0637</td>
<td>.0000</td>
<td>.0132</td>
<td>.0865</td>
<td>.0085</td>
<td>1.5911</td>
<td></td>
</tr>
<tr>
<td>6.7881</td>
<td>5.000</td>
<td>.0148</td>
<td>.0024</td>
<td>.0022</td>
<td>.0666</td>
<td>.0644</td>
<td>.0000</td>
<td>.0133</td>
<td>.0874</td>
<td>.0086</td>
<td>1.5925</td>
<td></td>
</tr>
<tr>
<td>6.8644</td>
<td>5.000</td>
<td>.0153</td>
<td>.0024</td>
<td>.0022</td>
<td>.0673</td>
<td>.0651</td>
<td>.0000</td>
<td>.0134</td>
<td>.0883</td>
<td>.0087</td>
<td>1.5939</td>
<td></td>
</tr>
<tr>
<td>6.9407</td>
<td>5.000</td>
<td>.0157</td>
<td>.0024</td>
<td>.0022</td>
<td>.0680</td>
<td>.0658</td>
<td>.0000</td>
<td>.0135</td>
<td>.0892</td>
<td>.0088</td>
<td>1.5953</td>
<td></td>
</tr>
<tr>
<td>7.0169</td>
<td>5.000</td>
<td>.0163</td>
<td>.0024</td>
<td>.0022</td>
<td>.0687</td>
<td>.0665</td>
<td>.0000</td>
<td>.0137</td>
<td>.0901</td>
<td>.0089</td>
<td>1.5967</td>
<td></td>
</tr>
<tr>
<td>7.0932</td>
<td>5.000</td>
<td>.0169</td>
<td>.0024</td>
<td>.0022</td>
<td>.0694</td>
<td>.0672</td>
<td>.0000</td>
<td>.0138</td>
<td>.0910</td>
<td>.0090</td>
<td>1.5981</td>
<td></td>
</tr>
<tr>
<td>7.1695</td>
<td>5.000</td>
<td>.0176</td>
<td>.0024</td>
<td>.0022</td>
<td>.0701</td>
<td>.0679</td>
<td>.0000</td>
<td>.0139</td>
<td>.0920</td>
<td>.0091</td>
<td>1.5995</td>
<td></td>
</tr>
<tr>
<td>7.2458</td>
<td>5.000</td>
<td>.0184</td>
<td>.0024</td>
<td>.0022</td>
<td>.0708</td>
<td>.0686</td>
<td>.0000</td>
<td>.0140</td>
<td>.0928</td>
<td>.0092</td>
<td>1.6009</td>
<td></td>
</tr>
<tr>
<td>7.3220</td>
<td>5.000</td>
<td>.0192</td>
<td>.0024</td>
<td>.0022</td>
<td>.0715</td>
<td>.0693</td>
<td>.0000</td>
<td>.0141</td>
<td>.0937</td>
<td>.0093</td>
<td>1.6023</td>
<td></td>
</tr>
<tr>
<td>7.3983</td>
<td>5.000</td>
<td>.0202</td>
<td>.0024</td>
<td>.0022</td>
<td>.0722</td>
<td>.0700</td>
<td>.0000</td>
<td>.0143</td>
<td>.0946</td>
<td>.0094</td>
<td>1.6037</td>
<td></td>
</tr>
<tr>
<td>7.4746</td>
<td>5.000</td>
<td>.0213</td>
<td>.0024</td>
<td>.0022</td>
<td>.0729</td>
<td>.0707</td>
<td>.0000</td>
<td>.0144</td>
<td>.0955</td>
<td>.0095</td>
<td>1.6051</td>
<td></td>
</tr>
<tr>
<td>7.5508</td>
<td>5.000</td>
<td>.0225</td>
<td>.0024</td>
<td>.0022</td>
<td>.0736</td>
<td>.0714</td>
<td>.0000</td>
<td>.0145</td>
<td>.0964</td>
<td>.0096</td>
<td>1.6065</td>
<td></td>
</tr>
<tr>
<td>7.6271</td>
<td>5.000</td>
<td>.0239</td>
<td>.0024</td>
<td>.0022</td>
<td>.0743</td>
<td>.0721</td>
<td>.0000</td>
<td>.0146</td>
<td>.0972</td>
<td>.0097</td>
<td>1.6079</td>
<td></td>
</tr>
<tr>
<td>7.7034</td>
<td>5.000</td>
<td>.0255</td>
<td>.0024</td>
<td>.0022</td>
<td>.0750</td>
<td>.0728</td>
<td>.0000</td>
<td>.0147</td>
<td>.0981</td>
<td>.0098</td>
<td>1.6093</td>
<td></td>
</tr>
<tr>
<td>7.7797</td>
<td>5.000</td>
<td>.0273</td>
<td>.0024</td>
<td>.0022</td>
<td>.0757</td>
<td>.0735</td>
<td>.0000</td>
<td>.0148</td>
<td>.0989</td>
<td>.0099</td>
<td>1.6107</td>
<td></td>
</tr>
<tr>
<td>7.8559</td>
<td>5.000</td>
<td>.0294</td>
<td>.0024</td>
<td>.0022</td>
<td>.0764</td>
<td>.0742</td>
<td>.0000</td>
<td>.0149</td>
<td>.0997</td>
<td>.0100</td>
<td>1.6122</td>
<td></td>
</tr>
<tr>
<td>7.9322</td>
<td>5.000</td>
<td>.0317</td>
<td>.0024</td>
<td>.0022</td>
<td>.0771</td>
<td>.0749</td>
<td>.0000</td>
<td>.0149</td>
<td>.1005</td>
<td>.0101</td>
<td>1.6136</td>
<td></td>
</tr>
</tbody>
</table>

(e) Page 5.

Figure 14.— Continued.
**TEST CASE** L/D=9 FOREBODY L/D=0.8 DB/D=0.51 CIRCULAR ARC NOZZLE NPR=5.09

**ITERATION NO 15**

MO = 8000  TT = 324.44 KELVIN  PT = 100720.0 PASCALS  L = 152400 METERS  SREF = 0.016242 80 METERS

BOUNDARY LAYER SEPARATION AT X/L = 9.529286

<table>
<thead>
<tr>
<th>X/L</th>
<th>Y/L</th>
<th>R/L</th>
<th>CP</th>
<th>CF</th>
<th>CDP</th>
<th>CDT</th>
<th>RD8/L</th>
<th>RC/L</th>
<th>DEL/L</th>
<th>DEL/L</th>
<th>THETA/L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.000</td>
<td>0.344</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.005</td>
<td>0.000</td>
<td>0.336</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.010</td>
<td>0.000</td>
<td>0.319</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.015</td>
<td>0.000</td>
<td>0.313</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.020</td>
<td>0.000</td>
<td>0.290</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.025</td>
<td>0.000</td>
<td>0.268</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.030</td>
<td>0.000</td>
<td>0.240</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.035</td>
<td>0.000</td>
<td>0.208</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.040</td>
<td>0.000</td>
<td>0.176</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.045</td>
<td>0.000</td>
<td>0.144</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.050</td>
<td>0.000</td>
<td>0.112</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.055</td>
<td>0.000</td>
<td>0.080</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.060</td>
<td>0.000</td>
<td>0.048</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.065</td>
<td>0.000</td>
<td>0.016</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.070</td>
<td>0.000</td>
<td>0.080</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.075</td>
<td>0.000</td>
<td>0.048</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
<tr>
<td>0.080</td>
<td>0.000</td>
<td>0.016</td>
<td>0.022</td>
<td>0.022</td>
<td>0.0776</td>
<td>0.0756</td>
<td>0.500</td>
<td>0.510</td>
<td>0.0140</td>
<td>0.1013</td>
<td>0.0095</td>
<td>1.5716</td>
</tr>
</tbody>
</table>

Figure 14.- Continued.
**Test Case**  L/D = 0.8  DB/D = 0.51  Circular Arc Nozzle  NPR = 5.03  Iteration No. 15

Mo = .8000  TT = 324.94 Kelvin  PT = 100720.0 Pascals  L = .192000 Meters  SRef = .018242 50 Meters

Boundary Layer Separation at X/L = 9.529286  Boundary Layer Reattachment at X/L = 11.633000

<table>
<thead>
<tr>
<th>X/L</th>
<th>R/L</th>
<th>CP</th>
<th>CF</th>
<th>CDP</th>
<th>CDF</th>
<th>CDT</th>
<th>RD/SL</th>
<th>RC/L</th>
<th>DEL/SL</th>
<th>DEL/L</th>
<th>THETA/L</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1000</td>
<td>.2840</td>
<td>.2065</td>
<td>.0011</td>
<td>=0006</td>
<td>.0946</td>
<td>.0940</td>
<td>.2858</td>
<td>.3469</td>
<td>.0552</td>
<td>.2312</td>
<td>.0309</td>
<td>1.7863</td>
</tr>
<tr>
<td>10.1500</td>
<td>.2839</td>
<td>.2083</td>
<td>.0011</td>
<td>=0006</td>
<td>.0947</td>
<td>.0941</td>
<td>.2835</td>
<td>.3422</td>
<td>.0556</td>
<td>.2340</td>
<td>.0313</td>
<td>1.7858</td>
</tr>
<tr>
<td>10.2500</td>
<td>.2850</td>
<td>.1863</td>
<td>.0012</td>
<td>=0006</td>
<td>.0949</td>
<td>.0943</td>
<td>.2849</td>
<td>.3366</td>
<td>.0523</td>
<td>.2317</td>
<td>.0299</td>
<td>1.7471</td>
</tr>
<tr>
<td>10.3000</td>
<td>.2857</td>
<td>.1034</td>
<td>.0013</td>
<td>=0006</td>
<td>.0950</td>
<td>.0944</td>
<td>.2853</td>
<td>.3349</td>
<td>.0492</td>
<td>.2292</td>
<td>.0286</td>
<td>1.7163</td>
</tr>
<tr>
<td>10.3500</td>
<td>.2862</td>
<td>.1458</td>
<td>.0013</td>
<td>=0006</td>
<td>.0952</td>
<td>.0946</td>
<td>.2859</td>
<td>.3334</td>
<td>.0470</td>
<td>.2276</td>
<td>.0277</td>
<td>1.6957</td>
</tr>
<tr>
<td>10.4000</td>
<td>.2866</td>
<td>.1322</td>
<td>.0014</td>
<td>=0006</td>
<td>.0953</td>
<td>.0947</td>
<td>.2864</td>
<td>.3322</td>
<td>.0455</td>
<td>.2266</td>
<td>.0271</td>
<td>1.6811</td>
</tr>
<tr>
<td>10.4500</td>
<td>.2869</td>
<td>.1201</td>
<td>.0014</td>
<td>=0006</td>
<td>.0956</td>
<td>.0948</td>
<td>.2868</td>
<td>.3312</td>
<td>.0443</td>
<td>.2259</td>
<td>.0265</td>
<td>1.6691</td>
</tr>
<tr>
<td>10.5000</td>
<td>.2872</td>
<td>.1085</td>
<td>.0015</td>
<td>=0006</td>
<td>.0956</td>
<td>.0948</td>
<td>.2872</td>
<td>.3303</td>
<td>.0431</td>
<td>.2252</td>
<td>.0260</td>
<td>1.6584</td>
</tr>
<tr>
<td>10.5500</td>
<td>.2875</td>
<td>.0981</td>
<td>.0015</td>
<td>=0006</td>
<td>.0957</td>
<td>.0951</td>
<td>.2875</td>
<td>.3297</td>
<td>.0422</td>
<td>.2248</td>
<td>.0256</td>
<td>1.6496</td>
</tr>
<tr>
<td>10.6000</td>
<td>.2878</td>
<td>.0888</td>
<td>.0015</td>
<td>=0006</td>
<td>.0957</td>
<td>.0952</td>
<td>.2877</td>
<td>.3292</td>
<td>.0414</td>
<td>.2244</td>
<td>.0252</td>
<td>1.6420</td>
</tr>
<tr>
<td>10.6500</td>
<td>.2881</td>
<td>.0785</td>
<td>.0015</td>
<td>=0006</td>
<td>.0960</td>
<td>.0954</td>
<td>.2881</td>
<td>.3286</td>
<td>.0405</td>
<td>.2241</td>
<td>.0248</td>
<td>1.6342</td>
</tr>
<tr>
<td>10.7000</td>
<td>.2884</td>
<td>.0686</td>
<td>.0016</td>
<td>=0006</td>
<td>.0962</td>
<td>.0956</td>
<td>.2883</td>
<td>.3282</td>
<td>.0398</td>
<td>.2240</td>
<td>.0245</td>
<td>1.6277</td>
</tr>
<tr>
<td>10.7500</td>
<td>.2886</td>
<td>.0581</td>
<td>.0016</td>
<td>=0006</td>
<td>.0964</td>
<td>.0958</td>
<td>.2886</td>
<td>.3279</td>
<td>.0393</td>
<td>.2234</td>
<td>.0242</td>
<td>1.6225</td>
</tr>
<tr>
<td>10.8000</td>
<td>.2888</td>
<td>.0487</td>
<td>.0016</td>
<td>=0006</td>
<td>.0966</td>
<td>.0960</td>
<td>.2888</td>
<td>.3276</td>
<td>.0386</td>
<td>.2228</td>
<td>.0240</td>
<td>1.6180</td>
</tr>
<tr>
<td>10.8500</td>
<td>.2889</td>
<td>.0395</td>
<td>.0016</td>
<td>=0006</td>
<td>.0968</td>
<td>.0962</td>
<td>.2889</td>
<td>.3274</td>
<td>.0389</td>
<td>.2224</td>
<td>.0236</td>
<td>1.6144</td>
</tr>
<tr>
<td>10.9000</td>
<td>.2891</td>
<td>.0300</td>
<td>.0016</td>
<td>=0006</td>
<td>.0970</td>
<td>.0964</td>
<td>.2901</td>
<td>.3272</td>
<td>.0381</td>
<td>.2221</td>
<td>.0233</td>
<td>1.6111</td>
</tr>
<tr>
<td>10.9500</td>
<td>.2892</td>
<td>.0407</td>
<td>.0016</td>
<td>=0006</td>
<td>.0973</td>
<td>.0966</td>
<td>.2902</td>
<td>.3271</td>
<td>.0379</td>
<td>.2220</td>
<td>.0236</td>
<td>1.6088</td>
</tr>
<tr>
<td>11.0000</td>
<td>.2893</td>
<td>.0368</td>
<td>.0016</td>
<td>=0006</td>
<td>.0974</td>
<td>.0968</td>
<td>.2903</td>
<td>.3270</td>
<td>.0377</td>
<td>.2233</td>
<td>.0239</td>
<td>1.6061</td>
</tr>
<tr>
<td>11.0500</td>
<td>.2894</td>
<td>.0335</td>
<td>.0017</td>
<td>=0006</td>
<td>.0976</td>
<td>.0970</td>
<td>.2904</td>
<td>.3269</td>
<td>.0375</td>
<td>.2257</td>
<td>.0234</td>
<td>1.6040</td>
</tr>
<tr>
<td>11.1000</td>
<td>.2895</td>
<td>.0306</td>
<td>.0017</td>
<td>=0006</td>
<td>.0978</td>
<td>.0972</td>
<td>.2905</td>
<td>.3269</td>
<td>.0374</td>
<td>.2262</td>
<td>.0233</td>
<td>1.6022</td>
</tr>
<tr>
<td>11.1500</td>
<td>.2896</td>
<td>.0277</td>
<td>.0017</td>
<td>=0006</td>
<td>.0980</td>
<td>.0974</td>
<td>.2906</td>
<td>.3269</td>
<td>.0373</td>
<td>.2266</td>
<td>.0233</td>
<td>1.6005</td>
</tr>
<tr>
<td>11.2500</td>
<td>.2897</td>
<td>.0232</td>
<td>.0017</td>
<td>=0006</td>
<td>.0985</td>
<td>.0978</td>
<td>.2907</td>
<td>.3268</td>
<td>.0371</td>
<td>.2276</td>
<td>.0232</td>
<td>1.5977</td>
</tr>
<tr>
<td>11.3000</td>
<td>.2898</td>
<td>.0211</td>
<td>.0017</td>
<td>=0006</td>
<td>.0987</td>
<td>.0981</td>
<td>.2908</td>
<td>.3268</td>
<td>.0370</td>
<td>.2281</td>
<td>.0232</td>
<td>1.5964</td>
</tr>
<tr>
<td>11.3500</td>
<td>.2898</td>
<td>.0191</td>
<td>.0017</td>
<td>=0006</td>
<td>.0989</td>
<td>.0983</td>
<td>.2908</td>
<td>.3268</td>
<td>.0369</td>
<td>.2286</td>
<td>.0232</td>
<td>1.5952</td>
</tr>
<tr>
<td>11.4000</td>
<td>.2899</td>
<td>.0171</td>
<td>.0017</td>
<td>=0006</td>
<td>.0991</td>
<td>.0985</td>
<td>.2909</td>
<td>.3268</td>
<td>.0369</td>
<td>.2290</td>
<td>.0232</td>
<td>1.5940</td>
</tr>
</tbody>
</table>

For Iteration 16, 8 iterations required for convergence in MIA2.

(g) Page 7.

Figure 14.- Continued.
POTENTIAL FLOW SOLUTION

**TEST CASE** L/D=9 FOREBODY L/D=0.6 DB/D=0.51 CIRCULAR ARC NOZZLE NPR=5.03

OFF-BODY UNIFORM AXISYMMETRIC FLOW

<table>
<thead>
<tr>
<th>X/L</th>
<th>R/L</th>
<th>VX</th>
<th>VR</th>
<th>VT</th>
<th>ETA</th>
<th>ML</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.800000</td>
<td>0.000000</td>
<td>1.062181</td>
<td>0.09934</td>
<td>1.06227</td>
<td>0.355666</td>
<td>0.856848</td>
</tr>
<tr>
<td>2</td>
<td>0.000000</td>
<td>0.000000</td>
<td>1.111666</td>
<td>0.03695</td>
<td>1.111723</td>
<td>0.811604</td>
<td>0.903119</td>
</tr>
<tr>
<td>3</td>
<td>0.200000</td>
<td>0.000000</td>
<td>1.119441</td>
<td>0.10927</td>
<td>1.249662</td>
<td>0.597425</td>
<td>0.315665</td>
</tr>
<tr>
<td>4</td>
<td>0.400000</td>
<td>0.000000</td>
<td>1.029521</td>
<td>1.147716</td>
<td>1.040064</td>
<td>0.168080</td>
<td>0.834438</td>
</tr>
<tr>
<td>5</td>
<td>0.600000</td>
<td>0.000000</td>
<td>0.956023</td>
<td>1.18833</td>
<td>0.963580</td>
<td>0.705490</td>
<td>0.767162</td>
</tr>
<tr>
<td>6</td>
<td>0.800000</td>
<td>0.000000</td>
<td>0.943930</td>
<td>0.82651</td>
<td>0.947542</td>
<td>0.504080</td>
<td>0.753125</td>
</tr>
<tr>
<td>7</td>
<td>0.800000</td>
<td>0.000000</td>
<td>0.937734</td>
<td>0.084026</td>
<td>0.941950</td>
<td>0.542331</td>
<td>0.748162</td>
</tr>
<tr>
<td>8</td>
<td>0.800000</td>
<td>0.000000</td>
<td>0.931333</td>
<td>0.095801</td>
<td>0.936049</td>
<td>0.5874312</td>
<td>0.742975</td>
</tr>
<tr>
<td>9</td>
<td>0.800000</td>
<td>0.000000</td>
<td>0.924388</td>
<td>0.102942</td>
<td>0.930082</td>
<td>0.635077</td>
<td>0.737722</td>
</tr>
<tr>
<td>10</td>
<td>0.800000</td>
<td>0.000000</td>
<td>0.917520</td>
<td>0.110636</td>
<td>0.924166</td>
<td>0.6875420</td>
<td>0.732524</td>
</tr>
</tbody>
</table>

(h) Page 8.

Figure 14.- Concluded.
A Neumann solution for inviscid external flow has been coupled to a modified Reshotko-Tucker integral boundary-layer technique, the control volume method of Presz for calculating flow in the separated region, and an inviscid one-dimensional solution for the jet exhaust flow in order to predict axisymmetric nozzle afterbody pressure distributions and drag. The viscous and inviscid flows are solved iteratively until convergence is obtained. A computer algorithm of this procedure has been written and is called DONBOL. This paper provides a description of the computer program and a guide to its use. Comparisons of the predictions of this method with experiment show that the method accurately predicts the pressure distributions of boattail afterbodies which have the jet exhaust flow simulated by solid bodies. For nozzle configurations which have the jet exhaust simulated by high-pressure air, the present method significantly underpredicts the magnitude of nozzle pressure drag. This deficiency results because the method neglects the effects of jet plume entrainment. This method is limited to subsonic free-stream Mach numbers below that for which the flow over the body of revolution becomes sonic.