REVISED FORTRAN PROGRAM FOR CALCULATING VELOCITIES AND STREAMLINES ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE

by Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1969
REVISED FORTRAN PROGRAM FOR CALCULATING VELOCITIES AND
STREAMLINES ON A BLADE-TO-BLADE STREAM SURFACE
OF A TURBOMACHINE

By Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 - CFSTI price $3.00
ABSTRACT

An existing FORTRAN IV computer program for blade-to-blade aerodynamic analysis of turbomachine blades was revised to obtain a simpler program consistent with related programs. The analysis is for two-dimensional, subsonic, compressible (or incompressible), nonviscous flow in a circular or straight infinite cascade of blades, which may be fixed or rotating. The flow may be axial, radial, or mixed, and the stream channel thickness may change in the through-flow direction. The results include streamline coordinates, velocity magnitude and direction throughout the passage, and the blade-surface velocities. This report includes a complete description of the input required by the program and the program listing.
REVISED FORTRAN PROGRAM FOR CALCULATING VELOCITIES AND STREAMLINES
ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE
by Theodore Katsanis and William D. McNally
Lewis Research Center

SUMMARY

An existing FORTRAN IV computer program for blade-to-blade aerodynamic analysis
of turbomachine blades was revised to obtain a simpler program consistent with related
programs. The analysis is for two-dimensional, subsonic, compressible (or incompressible),
nonviscous flow in a circular or straight infinite cascade of blades, which may be
fixed or rotating. The flow may be axial, radial, or mixed, and the stream channel
thickness may change in the through-flow direction.

The program input consists of blade and stream channel geometry, total flow condi-
tions, inlet and outlet flow angles, and blade-to-blade stream channel weight flow. The
output includes blade-surface velocities, velocity magnitude and direction at all interior
mesh points in the blade-to-blade passage, and streamline coordinates throughout the
passage.

This report includes a complete description of the input required by the program and
the program listing.

INTRODUCTION

In the design of blade rows for compressors and turbines, it is desirable to obtain
fluid velocities in the blade-to-blade passage and particularly on the blade surfaces. The
trend to highly loaded blading results in widely spaced blades with less of the passage
within a guided channel between blades. Stream filament techniques, applicable only
within guided channels, can therefore no longer be used to obtain velocities over the en-
tire blade surfaces. However, finite-difference methods can be used to obtain a solution
of the stream-function differential equation in both the guided and unguided portions of the
passage.

Computer programs have been written which generate coefficients for the finite-
difference equations, solve these equations, and differentiate the resulting values of stream function to obtain velocities throughout the blade-to-blade passage and on the blade surfaces. This was done in reference 1 for single blade row turbomachines or cascades, and in reference 2 for tandem or slotted blade machines.

When the program of reference 2 (TANDEM) was written, many improvements were made over the single blade row program (2DCP) of reference 1. This report describes a new program (TURBLE) which solves the same problem as 2DCP but incorporates all of the improvements of TANDEM. The coding in TURBLE is both simpler and more foolproof than that of 2DCP. Another reason for writing TURBLE has to do with the magnification program (MAGNFY) described in reference 3. The input to TURBLE has the same form as the input to TANDEM, and this simplifies the input coding to the magnification program (MAGNFY) of reference 3. It also allows the person using both TURBLE and TANDEM to put his input data in the same form in both cases. Further, TURBLE allows more interior mesh points in the solution region, and has its own error package independent of the Lewis computer system. Finally, the output of TURBLE has been expanded and clarified compared to the output of 2DCP.

Like 2DCP, TURBLE obtains the numerical solution for ideal, subsonic, compressible (or incompressible) flow for an axial-, radial-, or mixed-flow cascade of turbomachine blades. The cascade may be circular or straight (infinite), and may be fixed or rotating. The coordinates used are meridional streamline distance and angle in radians.

This report includes a complete description of input and program listing for TURBLE. The mathematical analysis, the detailed program procedure, and the program output for TURBLE are all very similar to that for TANDEM (ref. 2).

A TURBLE source deck on tape is available from COSMIC (Computer Software Management and Information Center), Computer Center, University of Georgia, Athens, Georgia 30601. The program number is COSMIC number LEW-10788.

SYMBOLS

\begin{align*}
m & \text{ meridional streamline distance, meters, see figs. 1 and 2} \\
r & \text{ radius from axis of rotation, meters} \\
s & \text{ angular blade spacing or pitch, rad} \\
V_\theta & \text{ tangential component of absolute fluid velocity, meters/sec} \\
W & \text{ fluid velocity relative to blade, meters/sec} \\
z & \text{ axial coordinate, meters} \\
\alpha & \text{ angle between meridional streamline and axis of rotation, rad, see fig. 1} \\
\beta & \text{ angle between relative velocity vector and meridional plane, rad, see fig. 1} \\
\end{align*}
\(\theta\) relative angular coordinate, rad, see fig. 1

\(\lambda\) prerotation \(rV_\theta_{in}\), meters^2/sec

\(\rho\) density, kg/meter^3

\(\omega\) rotational speed, rad/sec

Subscripts:
- \(cr\) critical velocity
- \(in\) inlet or upstream
- \(le\) leading edge
- \(out\) outlet or downstream
- \(te\) trailing edge

DESCRIPTION OF INPUT AND OUTPUT

The computer program requires as input a geometrical description in m-\(\theta\) coordinates of the blade surfaces, a description in m-\(r\) coordinates of the stream channel.

Figure 1. - Cylindrical coordinate system and velocity components.
Figure 2. - Blade-to-blade surface of revolution.

<table>
<thead>
<tr>
<th>1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
<th>65</th>
<th>70</th>
<th>75</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAM</td>
<td>AR</td>
<td>TIP</td>
<td>RHO1P</td>
<td>WITL</td>
<td>OMEGA</td>
<td>ORF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BETA1</td>
<td>BETA0</td>
<td>CHORD</td>
<td>STGR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M81</td>
<td>M80</td>
<td>MM</td>
<td>NBBI</td>
<td>NBL</td>
<td>NRSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>R01</td>
<td>BET1</td>
<td>BET01</td>
<td>SPLIT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSP1 ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THSP1 ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td>R02</td>
<td>BET12</td>
<td>BET02</td>
<td>SPLIT02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSP2 ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THSP2 ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MR ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSF ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BESP ARRAYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLDA1</td>
<td>ANNDK</td>
<td>EROR</td>
<td>STRF</td>
<td>SLCRD</td>
<td>INTVL</td>
<td>SURV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. - Input form. Card column numbers appear at top.
through the blades, appropriate gas constants, and operating conditions such as inlet
temperature and density, inlet and outlet flow angles, weight flow, and rotational speed.
Figures 1 and 2 show the m- \( \theta \) coordinate system for a typical blade-to-blade surface of revolution. Output obtained from the program includes velocity magnitude and direction at all interior mesh points in the blade-to-blade passage, blade-surface velocities, stream-function values throughout the blade-to-blade region of solution, and streamline locations.

Input

Figure 3 shows the input variables as they are punched on the data cards. There are two types of variables, geometric and nongeometric. The geometric input variables

![Figure 4](image1)  
**Figure 4.** - Geometric input variables on blade-to-blade solution region.

![Figure 5](image2)  
**Figure 5.** - Geometric input variables on a blade. BET1 and BET0 angles must be given as true angle \( \beta \), not as angles measured in m-\( \theta \) plane. Use \( \tan \beta = r \delta d \) to obtain \( \beta \), or measure true angle.
are shown in figures 4 to 6. **All input variables are described in the section which follows.** Further explanation of key variables is given in the section Instructions for Preparing Input.

The input variables are as follows:

- **GAM** specific heat ratio
- **AR** gas constant, joule/(kg)(°K)
- **TIP** inlet stagnation temperature, °K
- **RHOIP** inlet stagnation density, kg/meter$^3$
- **WTFL** mass flow per blade for the stream channel, kg/sec
- **OMEGA** rotational speed, $\omega$, rad/sec (Note that $\omega$ is negative if rotation is in the opposite direction of that shown in fig. 1)
- **ORF** value of overrelaxation factor to be used in the solution of the inner iteration simultaneous equations (If ORF = 0, the program calculates an estimated value for the overrelaxation factor. See p. 10 for discussion.)
- **BETAI** inlet flow angle $\beta_{le}$ along BG with respect to m-direction, deg (fig. 4)
- **BETAO** outlet flow angle $\beta_{te}$ along CF with respect to m-direction, deg (fig. 4)
- **CHORDF** overall length of blade in m-direction, meters (fig. 4)
- **STGRF** angular $\theta$-coordinate for center of trailing-edge circle of blade with respect to the center of leading-edge circle of blade, rad (fig. 4)
- **MBI** number of vertical mesh lines from AH to BG inclusive (fig. 4)
MBO number of vertical mesh lines from AH to CF inclusive (fig. 4)

MM total number of vertical mesh lines in the m-direction from AH to DE, maximum of 100 (fig. 4)

NBBI number of mesh spaces in θ-direction between AB and GH, maximum of 50 (fig. 4)

NBL number of blades

NRSP number of spline points for stream channel radius (RMSP) and thickness (BESP) coordinates, maximum of 50 (fig. 6)

RI1, RI2 leading-edge radii of the two blade surfaces, meters (fig. 5)

RO1, RO2 trailing-edge radii of the two blade surfaces, meters (fig. 5)

BETI1, BETI2 angles (with respect to m-direction) at tangent points of leading-edge radii with the two blade surfaces, deg (fig. 5) (These must be true angles in degrees. If angles are measured in the m-θ plane, i.e. dθ/dm, BETI1 and BETI2 can be obtained from the relation tan β = r(dθ/dm).)

BETO1, BETO2 angles (with respect to m-direction) at tangent points of trailing-edge radii with the two blade surfaces, deg (fig. 5) (These must also be true angles in degrees, like BETI1 and BETI2.)

SPLNO1, SPLNO2 number of blade spline points given for each surface as input, maximum of 50 (These include the first and last points (dummies) that are tangent to the leading- and trailing-edge radii (fig. 5).)

MSP1, MSP2 arrays of m-coordinates of spline points on the two blade surfaces, measured from the blade leading edge, meters (fig. 5) (The first and last points in each of these arrays can be blank or have a dummy value, since these points are calculated by the program. If blanks are used, and the last point is on a new card, a blank card must be used.)

THSP1, THSP2 arrays of θ-coordinates of spline points corresponding to MSP1 and MSP2, rad (fig. 5) (Dummy values are also used here in positions corresponding to those in MSP1 and MSP2.)

MR array of m-coordinates of spline points for the stream channel radii and the stream channel thicknesses, meters (fig. 6) (MR is measured from the leading edge of the blade. These coordinates should cover the entire distance from AH to DE, and may extend beyond these bounds. The total number of points is NRSP.)
array of \( r \)-coordinates of spline points for the stream channel radii, corresponding to the MR array, meters (fig. 6)

array of stream channel normal thicknesses corresponding to the MR and RMSP arrays, meters (fig. 6)

The remaining variables, starting with BLDAT, are used to indicate what output is desired. A value of zero for any of these variables will cause the output associated with that variable to be omitted. A value of 1 will cause the corresponding output to be printed for the final outer iteration only; 2, for the first and final iterations; and 3, for all outer iterations. Care should be used not to call for more output than is really useful. The following list gives the output associated with each of these variables.

BLDAT all geometrical information which does not change from iteration to iteration (i.e., coordinates and first and second derivatives of all blade surface spline points; blade coordinates and blade slopes where vertical mesh lines meet each blade surface; radii and stream channel thicknesses corresponding to each vertical mesh line; \( m \)-coordinate, stream channel radius and thickness, and blade surface angles and slopes where horizontal mesh lines intersect each blade; and ITV and IV arrays (internal variables describing the location of the blade surfaces with respect to the finite difference grid).)

AANDK the coefficient array, the constant vector, and the indexes of all adjacent points for each point in the finite-difference mesh (This information is needed for debugging the program only.)

ERSOR the maximum change in the stream function at any point for each iteration of the SOR equation, eq. (A8), ref. 2

STRFN value of the stream function at each unknown mesh point in the region

SLCRD streamline \( \theta \)-coordinates at each vertical mesh line, and streamline plot

INTVL velocity and flow angle at each interior mesh point

SURVL \( m \)-coordinate, surface velocity, flow angle, distance along surface, and \( W/W_{cr} \) based on meridional velocity components where each vertical mesh line meets each blade surface; \( m \)-coordinate, surface velocity, flow angle, distance along surface, and \( W/W_{cr} \) based on tangential velocity components where each horizontal mesh line meets each blade surface; and plot of blade-surface velocities against meridional streamline distance, meters. (It is suggested that SURVL=3 be used. This will give surface velocities after each outer iteration, so that satisfactory velocities may be obtained even when final convergence is not reached.)
Instructions for Preparing Input

Units of measurement. - The International System of Units (ref. 4) is used throughout this report. However, the program does not use any constants which depend on the system of units being used. Therefore, any consistent set of units may be used in preparing input for the program. For example, if force, length, temperature, and time are chosen independently, mass units are obtained from force = mass x acceleration. The gas constant $R$ must then have the units of force times length divided by mass times temperature (energy per unit mass per degree temperature). Density is mass per unit volume, and weight flow is mass per unit time. Output then gives velocity in the chosen units of length per unit time. Since any consistent set of units can be employed, the output is not labeled with any units.

Blade and stream channel geometry. - The upper and lower surfaces of the blade are each defined by specifying three things: leading- and trailing-edge radii, angles at which these radii are tangent to the blade surfaces, and m- and $\theta$-coordinates of several points along each surface. These angles and coordinates are used to define a cubic spline curve fit (ref. 5) to the surface. The standard sign convention is used for angles, as indicated in figure 5.

A cubic spline curve is a piecewise cubic polynomial which expresses mathematically the shape taken by an idealized spline passing through the given points. Reference 5 describes a method for determining the equation of the spline curve. Using this method, few points are required to specify most blade shapes accurately, usually no more than five or six, in addition to the two end points. As a guide, enough points should be specified so that a physical spline passing through these points would accurately follow the blade shape. This means that the spline points should be closer where there is large curvature and farther apart where there is small curvature.

The coordinates for either surface of the blade are given with respect to the leading edge, with the leading edge of the blade being defined as the furthest point upstream.

The mean stream surface of revolution (as seen in the meridional plane, fig. 6) and the stream channel thickness are also fitted with cubic spline curves. The m-coordinates for the mean stream surface are independent of the m-coordinates for blade surfaces.

Inlet and outlet flow angles. - The values of $\beta_{le}$ and $\beta_{te}$ are given as average values on BG and CF, respectively. If the flow is axial these flow angles are the same as the flow angles at AH and DE. If flow is radial or mixed, and these angles are not known on BG and CF, $\beta_{le}$ and $\beta_{te}$ must be calculated by equation (B15) of reference 1 or equation (B14) of reference 2.

Defining the mesh. - A finite-difference mesh is used for the solution of the basic differential equation. A typical mesh pattern is shown in figure 7. The mesh spacing and the extent of the upstream and downstream regions are determined by the values of MBI,
MBO, and MM of the input. The mesh spacing must be chosen so that there are not more than 2500 unknown mesh points.

Values of MBI, MBO, and MM should be determined so that the mesh which results has blocks which are approximately square. To achieve this, a value for NBBI is first chosen arbitrarily (15 to 20 is typical). NBBI is the number of mesh spaces spanning the blade pitch, $s$, where $s = 2\pi/NBL$. Dividing $s$ by NBBI gives the mesh spacing, $HT$, in the $\theta$-direction in radians. Multiplying $HT$ by an average radius (RMSP) of the stream channel gives an average value for the actual mesh spacing in the $\theta$-direction. The value of CHORD should then be used with this tangential mesh spacing to calculate the approximate number of mesh spaces along the blade in the $m$-direction. This will give MBO once MBI is chosen. Generally, MBI is given a value of $10$. MM, likewise, is usually given a value $10$ more than MBO.

Overrelaxation factor. - ORF is the overrelaxation factor used in each inner iteration in the solution of the simultaneous finite difference equations. (See ref. 2, p. 101).
ORF may be set to zero, or some value between 1 and 2. ORF is usually given as zero for the initial run of a given blade geometry and mesh spacing (MBI, NBI, etc.). In this case the program uses extra time and calculates an optimum value for ORF. It does this by means of an iterative process, and on each iteration the current estimate of the optimum value for ORF is printed. The final estimate is the one used by the program for ORF. If the user does not change the mesh indexes MBI, MBO, MM, and NBI between runs, even though blade geometry or other input does change, he may use this final estimate of ORF in the input, saving the time used in its computation. In all cases, if ORF is not zero, it should have a value greater than 1 and less than 2.

Actually, the value of ORF is not as critical as the user might think. It gets more critical as the optimum value gets close to 2. For any run of a given set of data, only small changes will occur in the rate of convergence in SOR as long as the difference is not zero, it should have a value greater than 1 and less than 2.

actually, the value of ORF is not as critical as the user might think. It gets more critical as the optimum value gets close to 2. For any run of a given set of data, only small changes will occur in the rate of convergence in SOR as long as the difference is not zero, it should have a value greater than 1 and less than 2.

Format for input data. - All the numbers on the card beginning with MBI and on the card beginning with BLDAT are integers (no decimal point) in a 5-column field (see fig. 3). These must all be right adjusted. The input variables on all other data cards are real numbers (punch decimal point) in a 10-column field.

Incompressible flow. - While the program is written for compressible flow, it can be easily used for incompressible flow. To do so specify GAM = 1.5, AR = 1000, and TIP = 10^6 as input. This results in a single outer iteration of the program to obtain the stream function solution.

Straight infinite cascade. - The program is as easily applied to straight infinite cascades as circular cascades. Since the radius and number of blades (NBL) for such a cascade would actually be infinite, an artificial convention must be adopted. The user should pick a value for NBL, for instance 20 or 30. Then, since the blade pitch sr (fig. 4) is known, an artificial radius can be computed from

\[ r = \frac{NBL \times (sr)}{2\pi} \]

This \( r \) should be used to compute the \( \theta \)-coordinates requires as input (THSP1, THSP2, and STGR) by dividing coordinates in the tangential direction by \( r \).

Axial flow. - For a two-dimensional cascade with constant stream channel thickness, constant values should be given for the MR, RMSP, and BESP arrays. Only two points are required for each of these arrays in this case. The two values of MR should be chosen so that they are further upstream and downstream than the boundaries AH and DE. The two values of RMSP and BESP should equal the constants \( r \) and \( b \).
<table>
<thead>
<tr>
<th>TABLE I - SAMPLE OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AXIAL STATOR - MEAN SECTION - COMPRESSIBLE</strong></td>
</tr>
<tr>
<td>GAM</td>
</tr>
<tr>
<td>1.4000000</td>
</tr>
<tr>
<td>Beta</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>NBl</td>
</tr>
<tr>
<td>-15</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 1 -- UPPER SURFACE**

<table>
<thead>
<tr>
<th>Blad</th>
<th>ArW</th>
<th>Bet1</th>
<th>Bet1</th>
<th>Spn1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3810000E-02</td>
<td>0.8900000E-03</td>
<td>28.300000</td>
<td>-72.400000</td>
<td>7.000000</td>
</tr>
<tr>
<td>0</td>
<td>0.8575000E-02</td>
<td>0.1715000E-01</td>
<td>0.2572500E-01</td>
<td>0.3430000E-01</td>
</tr>
<tr>
<td>0</td>
<td>0.1769000E-01</td>
<td>0.1538000E-01</td>
<td>-0.5100000E-02</td>
<td>-0.4654000E-01</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 2 -- LOWER SURFACE**

<table>
<thead>
<tr>
<th>Blad</th>
<th>ArW</th>
<th>Bet2</th>
<th>Bet2</th>
<th>Spn1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3810000E-02</td>
<td>0.8900000E-03</td>
<td>-14.200000</td>
<td>-56.100000</td>
<td>6.000000</td>
</tr>
<tr>
<td>0</td>
<td>0.8575000E-02</td>
<td>0.1715000E-01</td>
<td>0.2572500E-01</td>
<td>0.3430000E-01</td>
</tr>
<tr>
<td>0</td>
<td>-0.1562000E-01</td>
<td>-0.2854000E-01</td>
<td>-0.5070000E-01</td>
<td>-0.8250000E-01</td>
</tr>
</tbody>
</table>

**MR ARRAY**

<table>
<thead>
<tr>
<th>Blad</th>
<th>ArW</th>
<th>Bet2</th>
<th>Bet2</th>
<th>Spn1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.5000000E-01</td>
<td>0.1000000</td>
<td>-0.3302000</td>
<td>-0.1016000</td>
<td></td>
</tr>
</tbody>
</table>

**BLOAD aAND OR ON ERROR STRFN SLCRD INTVL SALT**

<table>
<thead>
<tr>
<th>Blad</th>
<th>ArW</th>
<th>Bet2</th>
<th>Bet2</th>
<th>Spn1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**BLADE DATA AT INPUT SPLINE POINTS**

<table>
<thead>
<tr>
<th>BLADE SURFACE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong></td>
</tr>
<tr>
<td>0.20037E-02</td>
</tr>
<tr>
<td>0.89750E-02</td>
</tr>
<tr>
<td>0.17150E-01</td>
</tr>
<tr>
<td>0.25725E-01</td>
</tr>
<tr>
<td>0.34300E-01</td>
</tr>
<tr>
<td>0.28080E-01</td>
</tr>
<tr>
<td>0.42800E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLADE SURFACE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong></td>
</tr>
<tr>
<td>0.28754E-02</td>
</tr>
<tr>
<td>0.89750E-02</td>
</tr>
<tr>
<td>0.17150E-01</td>
</tr>
<tr>
<td>0.25725E-01</td>
</tr>
<tr>
<td>0.34300E-01</td>
</tr>
<tr>
<td>0.41023E-01</td>
</tr>
</tbody>
</table>
### Freestream Data

<table>
<thead>
<tr>
<th>Leading Edge B-G</th>
<th>Maximum Value</th>
<th>Critical Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.9384</td>
<td>241.234</td>
<td>110.645</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TRailing Edge C-F</th>
<th>Maximum Value</th>
<th>Critical Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>180.007</td>
<td>241.234</td>
<td>110.645</td>
</tr>
</tbody>
</table>

### Calculated Program Constants

- **PITCH**: 0.1256637
- **HT**: 0.6283185E-02
- **HNL**: 0.2508824E-02
- **ITMIN**: -17
- **ITMAX**: 19
- **LAMBDA**: 0

### Number of Interior Mesh Points

- 846

### Surface Boundary Values

<table>
<thead>
<tr>
<th>Surface</th>
<th>BV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

### Blade Data at Intersections of Vertical Mesh Lines with Blades

<table>
<thead>
<tr>
<th>M</th>
<th>BLADE SURFACE 1</th>
<th>BLADE SURFACE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TV</td>
<td>DTMOV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.100000E+11</td>
</tr>
<tr>
<td>0.25088E-02</td>
<td>0.10966E-01</td>
<td>1.56401</td>
</tr>
<tr>
<td>0.30176E-02</td>
<td>0.14447E-01</td>
<td>1.20260</td>
</tr>
<tr>
<td>0.75658E-02</td>
<td>0.16958E-01</td>
<td>0.79070</td>
</tr>
<tr>
<td>0.10000E-01</td>
<td>0.18373E-01</td>
<td>0.32903</td>
</tr>
</tbody>
</table>

### Stream Sheet Coordinates and Thickness Table

<table>
<thead>
<tr>
<th>M</th>
<th>R</th>
<th>DB/DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.35124E-01</td>
<td>0.33020</td>
</tr>
<tr>
<td>2</td>
<td>0.32615E-01</td>
<td>0.33020</td>
</tr>
<tr>
<td>3</td>
<td>0.30106E-01</td>
<td>0.33020</td>
</tr>
<tr>
<td>4</td>
<td>0.27597E-01</td>
<td>0.33020</td>
</tr>
<tr>
<td>5</td>
<td>0.25088E-01</td>
<td>0.33020</td>
</tr>
</tbody>
</table>

### IV Array

<table>
<thead>
<tr>
<th>M</th>
<th>IV ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>61</td>
</tr>
<tr>
<td>5</td>
<td>81</td>
</tr>
</tbody>
</table>

### IVIV Array

<table>
<thead>
<tr>
<th>BLADE SURFACE</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>19</td>
</tr>
</tbody>
</table>
### TABLE I. - Continued. SAMPLE OUTPUT

#### M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINES WITH BLADE

**MH ARRAY - BLADE SURFACE 1**

<table>
<thead>
<tr>
<th>MH</th>
<th>RMH</th>
<th>DEH</th>
<th>BETAH</th>
<th>DTOMH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3302</td>
<td>0.1016</td>
<td>90.000</td>
<td>0.1000E 11</td>
</tr>
<tr>
<td>0.6132E-03</td>
<td>0.3302</td>
<td>0.1016</td>
<td>57.041</td>
<td>4.6708</td>
</tr>
<tr>
<td>0.3582E-02</td>
<td>0.3302</td>
<td>0.1016</td>
<td>25.092</td>
<td>1.4156</td>
</tr>
<tr>
<td>0.1906E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-29.610</td>
<td>-1.7211</td>
</tr>
<tr>
<td>0.2201E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-60.095</td>
<td>-2.9498</td>
</tr>
<tr>
<td>0.2420E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-46.766</td>
<td>-3.2212</td>
</tr>
</tbody>
</table>

**MH ARRAY - BLADE SURFACE 2**

<table>
<thead>
<tr>
<th>MH</th>
<th>RMH</th>
<th>DEH</th>
<th>BETAH</th>
<th>DTOMH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4152E-03</td>
<td>0.3302</td>
<td>0.1016</td>
<td>57.041</td>
<td>4.6708</td>
</tr>
<tr>
<td>0.4759E-02</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-13.294</td>
<td>-0.7156</td>
</tr>
<tr>
<td>0.1140E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-23.596</td>
<td>-1.3228</td>
</tr>
<tr>
<td>0.1539E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-31.073</td>
<td>-1.8250</td>
</tr>
<tr>
<td>0.1850E-01</td>
<td>0.3302</td>
<td>0.1016</td>
<td>-36.109</td>
<td>-2.2140</td>
</tr>
</tbody>
</table>

#### THETA COORDINATES OF HORIZONTAL MESH LINES

<table>
<thead>
<tr>
<th>IT</th>
<th>THETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>-17</td>
<td>-0.10681</td>
</tr>
<tr>
<td>-16</td>
<td>-0.10053</td>
</tr>
<tr>
<td>-15</td>
<td>-0.94246E-01</td>
</tr>
<tr>
<td>-14</td>
<td>-0.87965E-01</td>
</tr>
<tr>
<td>-13</td>
<td>-0.81681E-01</td>
</tr>
</tbody>
</table>

#### IP I P1 I P2 I P3 I P4 A(I) A(2) A(3) A(4) K

<table>
<thead>
<tr>
<th>IM</th>
<th>IT</th>
<th>IT1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ESTIMATED OPTIMUM ORF = 2.000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESTIMATED OPTIMUM ORF = 1.999756</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESTIMATED OPTIMUM ORF = 1.999655</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESTIMATED OPTIMUM ORF = 1.999655</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESTIMATED OPTIMUM ORF = 1.999655</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ERROR = 1.72602609</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR = 1.46492096</td>
</tr>
<tr>
<td>ERROR = 1.58245170</td>
</tr>
<tr>
<td>ERROR = 1.51499141</td>
</tr>
<tr>
<td>ERROR = 1.53113121</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STREAM FUNCTION VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM = 1</td>
</tr>
<tr>
<td>IM = 2</td>
</tr>
<tr>
<td>IM = 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME = 1.7294 MIN.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STREAMLINE COORDINATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>M COORDINATE</td>
</tr>
<tr>
<td>-0.3512353E-01</td>
</tr>
<tr>
<td>-0.3261671E-01</td>
</tr>
<tr>
<td>-0.3010588E-01</td>
</tr>
</tbody>
</table>
STREAMLINES ARE PLOTTED WITH THETA ACROSS THE PAGE AND M DOWN THE PAGE
### Velocities at Interior Mesh Points

<table>
<thead>
<tr>
<th>IM = 1</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.703</td>
<td>0.01</td>
<td>61.750</td>
<td>-0.02</td>
<td>61.818</td>
<td>-0.03</td>
<td>61.898</td>
<td>-0.03</td>
<td>61.971</td>
<td>-0.03</td>
<td>62.049</td>
<td>0.00</td>
<td>62.149</td>
</tr>
<tr>
<td>62.173</td>
<td>0.01</td>
<td>62.125</td>
<td>0.02</td>
<td>62.175</td>
<td>-0.01</td>
<td>62.209</td>
<td>0.03</td>
<td>62.281</td>
<td>0.03</td>
<td>62.319</td>
<td>0.01</td>
<td>62.419</td>
</tr>
<tr>
<td>61.819</td>
<td>0.03</td>
<td>61.751</td>
<td>0.02</td>
<td>61.702</td>
<td>0.01</td>
<td>61.676</td>
<td>0.01</td>
<td>61.678</td>
<td>-0.01</td>
<td>61.678</td>
<td>-0.01</td>
<td>61.678</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IM = 2</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.703</td>
<td>0.02</td>
<td>61.750</td>
<td>0.03</td>
<td>61.818</td>
<td>0.04</td>
<td>61.898</td>
<td>0.04</td>
<td>61.971</td>
<td>0.04</td>
<td>62.049</td>
<td>0.01</td>
<td>62.149</td>
</tr>
<tr>
<td>62.060</td>
<td>0.04</td>
<td>62.127</td>
<td>0.03</td>
<td>62.175</td>
<td>0.02</td>
<td>62.209</td>
<td>0.01</td>
<td>62.281</td>
<td>0.01</td>
<td>62.319</td>
<td>0.01</td>
<td>62.419</td>
</tr>
<tr>
<td>62.173</td>
<td>-0.02</td>
<td>62.125</td>
<td>-0.03</td>
<td>62.059</td>
<td>-0.04</td>
<td>61.980</td>
<td>-0.04</td>
<td>61.898</td>
<td>-0.04</td>
<td>61.898</td>
<td>-0.04</td>
<td>61.898</td>
</tr>
<tr>
<td>61.819</td>
<td>-0.04</td>
<td>61.751</td>
<td>-0.03</td>
<td>61.703</td>
<td>-0.02</td>
<td>61.676</td>
<td>-0.01</td>
<td>61.678</td>
<td>-0.01</td>
<td>61.678</td>
<td>-0.01</td>
<td>61.678</td>
</tr>
</tbody>
</table>

### Iteration No. 1

**Maximum Relative Change in Density**: 0.5774

### Surface Velocities Based on Meridional Components

<table>
<thead>
<tr>
<th>M</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>SURF. LENGTH</th>
<th>W/WCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>90.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.250E-02</td>
<td>99.666</td>
<td>27.31</td>
<td>0.4405E-02</td>
<td>0.315</td>
</tr>
<tr>
<td>0.501E-02</td>
<td>114.60</td>
<td>21.66</td>
<td>0.7165E-02</td>
<td>0.369</td>
</tr>
<tr>
<td>0.752E-02</td>
<td>134.41</td>
<td>14.63</td>
<td>0.9807E-02</td>
<td>0.432</td>
</tr>
<tr>
<td>0.100E-01</td>
<td>158.08</td>
<td>6.20</td>
<td>0.1236E-01</td>
<td>0.509</td>
</tr>
</tbody>
</table>

### Surface Velocities Based on Tangential Components

<table>
<thead>
<tr>
<th>M</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>SURF. LENGTH</th>
<th>W/WCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11.357</td>
<td>90.00</td>
<td>0.3656E-01</td>
<td></td>
</tr>
<tr>
<td>0.613E-03</td>
<td>76.293</td>
<td>57.04</td>
<td>0.2476</td>
<td></td>
</tr>
<tr>
<td>0.358E-02</td>
<td>100.30</td>
<td>25.05</td>
<td>0.3229</td>
<td></td>
</tr>
<tr>
<td>0.190E-01</td>
<td>240.79</td>
<td>-29.61</td>
<td>0.7751</td>
<td></td>
</tr>
<tr>
<td>0.220E-01</td>
<td>260.01</td>
<td>-40.10</td>
<td>0.7726</td>
<td></td>
</tr>
</tbody>
</table>

### Blade Surface 1

<table>
<thead>
<tr>
<th>M</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>W/WCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>47.792</td>
<td>-57.04</td>
<td>0.1518</td>
</tr>
<tr>
<td>0.475E-02</td>
<td>48.156</td>
<td>-13.29</td>
<td>0.1550</td>
</tr>
<tr>
<td>0.114E-01</td>
<td>57.949</td>
<td>-23.60</td>
<td>0.1865</td>
</tr>
<tr>
<td>0.153E-01</td>
<td>62.955</td>
<td>-31.07</td>
<td>0.2027</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>VELOCITY (m/s)</th>
<th>ANGLE (deg)</th>
<th>W/WCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>76.293</td>
<td>57.04</td>
<td>0.2476</td>
</tr>
<tr>
<td>0.613E-03</td>
<td>100.30</td>
<td>25.05</td>
<td>0.3229</td>
</tr>
<tr>
<td>0.190E-01</td>
<td>240.79</td>
<td>-29.61</td>
<td>0.7751</td>
</tr>
<tr>
<td>0.220E-01</td>
<td>260.01</td>
<td>-40.10</td>
<td>0.7726</td>
</tr>
</tbody>
</table>
### TABLE I - Concluded. SAMPLE OUTPUT

**BLADE SURFACE VELOCITIES**

<table>
<thead>
<tr>
<th>Velocity (m/s)</th>
<th>0.0</th>
<th>0.01</th>
<th>0.02</th>
<th>0.03</th>
<th>0.04</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>0.01</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>0.02</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>0.03</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>0.04</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>0.05</td>
<td>0</td>
<td>+</td>
<td>X</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

**VELOCITY (W) VS. MERIDIONAL STREAMLINE DISTANCE (X) DOWN THE PAGE**

- + - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT
- * - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT
- X - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT
- 0 - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT

16 TIME = 1.0575 MIN.
Sample output is given in table I for the axial-flow stator example of reference 1. The blade shape is shown in figure 7. Since the complete output would be lengthy, only the first few lines of each section of output are reproduced herein. Most of the output is optional, and is controlled by the final input card, as already described. In some instances output labels are simply internal variable names.

Each section of the sample output in table I has been numbered to correspond to the following description:

1. The first output is a listing of the input data. All items are labeled as on the input form (fig. 3).

2. This is the output corresponding to BLDAT. (See the list of input variables. See ref. 2 for meaning of undefined labels.)

3. The relative free-stream velocity \( W \); the relative critical velocity \( W_{cr} \); and the maximum value of the mass flow parameter \( \rho W \) (corresponding to \( W = W_{cr} \)) are given at the leading edge of the blade (BG) and the trailing edge of the blade (CF). The inlet (outlet) free-stream flow angle \( \beta_{in} \) (\( \beta_{out} \)) at boundary AH (DE) is given. These angles are based on the input angles \( \beta_{in} \), \( \beta_{out} \), \( \beta_{in} \), and \( \beta_{out} \).

4. These are calculated program constants, including the pitch from blade to blade, the mesh spacing, the minimum and maximum values of IT in the solution region (ITMIN and ITMAX), and the value of the prewhirl \( \lambda \) (eq. (B8), ref. 2).

5. This is the number of mesh points in the entire solution region at which the stream function is unknown.

6. This is the boundary value (BV) of the stream function on each of the blade surfaces.

7. This is the output corresponding to AANDK.

8. If the program calculates an optimum overrelaxation factor (i.e., \( \text{ORF} = 0 \) in the input), then the successive estimates to the optimum value of \( \text{ORF} \) are printed. The last printed value of the estimated optimum \( \text{ORF} \) is the value of the overrelaxation factor (ORF) used by the program.

9. This is the output corresponding to ERSOR.

10. This is the output corresponding to STRFN.

11. This is the total execution time after obtaining the stream function solution for each outer iteration.

12. This is the output corresponding to SLCRD.

13. This is the output corresponding to INTVL.

14. This gives the maximum relative change in the density, for each outer iteration.

15. This is the output corresponding to SURVL.
(16) This is the total execution time after all calculations are completed for an outer iteration.

**ERROR CONDITIONS**

(1) SPLINT USED FOR EXTRAPOLATION  
**EXTRAPOLATED VALUE = X.XXX**  
SPLINT is normally used for interpolation, but may be used for extrapolation in some cases. When this occurs, the above message is printed as well as the input and output of SPLINT. Calculations proceed normally after this printout.

(2) BLCD CALL NO. XX  
**M COORDINATE IS NOT WITHIN BLADE**  
This message is printed by subroutine BLCD if the m-coordinate given this subroutine as input is not within the bounds of the blade surface for which BLCD is called. The value of m and the blade-surface number are also printed when this happens. This may be caused by an error in the integer input items for the program.

The location of the error in the main program is given by means of BLCD CALL NO. XX, which corresponds to locations noted by comment cards at each MHORIZ, ROOT, and BLCD call in the program.

(3) ROOT CALL NO. XX  
**ROOT HAS FAILED TO CONVERGE IN 1000 ITERATIONS**  
This message is printed by subroutine ROOT if a root cannot be located. The input to ROOT is also printed. The user should thoroughly check the input to the main program.

The location of the error in the main program is given by means of ROOT CALL NO. XX, which corresponds to locations noted by comment cards at each MHORIZ and ROOT call in the program.

(4) DENSTY CALL NO. XX  
**NER(1) = XX**  
**RHO*W IS X.XXXX TIMES THE MAXIMUM VALUE FOR RHO*W**  
This message is printed if the value of $\rho W$ at some mesh point is so large that there is no solution for the value of $\rho$ and $W$. This indicates a locally supersonic condition, which can be eliminated by decreasing WTFL in the input.

If RHO*W is too large, TURBLE still attempts to calculate a solution. This often permits an approximate solution to be obtained, which is valid at all the subsonic points in the region. In other cases the value of $W$ is reduced at some of the points in question during later iterations, resulting in a valid final solution for these points. The program counts the number of times supersonic flow has been located at any point during a given run (NER(1)). When NER(1) = 50, the program is stopped.
The location of the error in the main program is given by means of DENSTY CALL NO. XX, which corresponds to locations noted by comment cards at each DENSTY call in the program.

(5) MM, NBBI, NRSP, OR SOME SPLNO IS TOO LARGE
If this is printed, reduce the appropriate inputs to their allotted maximum values.

(6) WTFL IS TOO LARGE AT BLADE LEADING EDGE
This is printed if WTFL is greater than the choking mass flow for the boundary BG. If this message is printed, WTFL is cut in half by the program and calculations proceed as usual.

(7) ONE OF THE MH ARRAYS IS TOO LARGE
This is printed if there are more than 100 intersections of horizontal mesh lines with any blade surface. In this case NBBI should be reduced.

(8) THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500
This is printed if there are more than the allowable number of finite-difference grid points. Either MM or NBBI must be reduced.

(9) SEARCH CANNOT FIND M IN THE MH ARRAY.
If this is printed, the value of m and the blade-surface number are also printed. The user should thoroughly check the input to the main program.

PROGRAM LISTING

The program is identical to TANDEM (ref. 2) except for deleting all code dealing with the rear blade and making necessary corrections to the remainder. The program procedure for TANDEM, given in reference 2, is applicable also for TURBLE, except for small deletions. Also, the FORTRAN dictionary for TANDEM is valid for TURBLE.

```fortran
COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON /AUKRHO/ A(2500, 4), U(2500), K(2500), RHO(2500)
COMMON /INP/GAM, AR, TIP, RHODIP, WTFL, OMEGA, ORF, BETAI, BETAO,
1 MB1, MB0, MM, NBBI, NBL, NRSP, MR(50), RMSP(50), BESP(50),
2 BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/MBIM1, MBIP1, MBDM1, MBOP1, MM1, HM1, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, TWM, CTIP, TCRG, TBI, TBO, LAMBDA, TWL, ITMIN, ITMAX,
2 NIP, IMS(2), BV(2), MV(100), ITV(101), ITV(100, 2), TV(100, 2),
3 DTMV(100, 2), BETAV(100, 2), MH(100, 2), DTMH(100, 2), BETAH(100, 2),
4 RMH(100, 2), BEH(100, 2), RM(100), BE(100), DBDM(100), SAL(100),
5 AAAI(100)
COMMON /GEOMIN/ CHJRDI(2), STGR(2), MLE(2), THLE(2), RMI(2), RM0(2),
1 RI(2), RO(2), BETI(2), BETO(2), NSPI(2), MSP(50, 2), THSP(50, 2)
COMMON /RHOS/RH0H(100, 2), RHUV(100, 2)
COMMON /BLDCM/ EM(50, 2), INIT(2)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SLCRD, SURVL, ATEMP, SURF, FIRST,
1 UPPER, SL, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MM, MLE, MR, MSL, MSP, MV, TMVIM
CALL TIME1(T1)
```
10 IEND = -1
ITER = 0
INIT(1) = 0
INIT(2) = 0
CALL INPUT
CALL PRECAL
30 CALL COEF
CALL SOR
CALL TIME(T2)
TIME = (T2 - T1) / 3600.
WRITE(6, 1000) TIME
CALL SLAX
CALL TANG
CALL VELOCITY
CALL TIME(T2)
TIME = (T2 - T1) / 3600.
WRITE(6, 1000) TIME
IF (NER(2).GT.0) GO TO 10
IF (IEND) 30, 30, 10
1000 FORMAT (8HLTIME = ,F7.4, 5H MIN.)
END

SUBROUTINE INPUT
C INPUT READS AND PRINTS ALL INPUT DATA CARDS AND CALCULATES HORIZONTAL SPACING (MV ARRAY)
C
COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON IAIKRHO, A(250), U(250), K(250), RHO(250)
COMMON /INPGAM, AR, TIP, RHOIP, WTFL, OMEGA, ORF, BETA, BETA0,
1 MB, MBO, MM, NBB, NBL, NRS, MR(50), RMSP(50), RESP(50),
2 BLDATA, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/MBI, MBIP1, MMIP1, MBOP1, MMM1, HM, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, IWW, CPTIP, TGROUP, TBI, TBO, LAMBDA, TWL, ITMIN, ITMAX,
2 NIP, IMS(2), BV(2), MV(100), TV(101), TV(100, 2), TV(100, 2),
3 DTDV(100, 2), BETAV(100, 2), MH(100, 2), DTD(100, 2), BETAM(100, 2),
4 RMH(100, 2), BEM(100, 2), RM(100), BE(100), DRDM(100), SAI(100),
5 AAA(100)
COMMON /GEOMIN/ CHORD(2), STGR(2), MLET(2), THLE(2), RMI(2), RMO(2),
1 RI(2), RD(2), BETA(2), BETO(2), NSPI(2), MSP(50, 2), THSP(50, 2)
COMMON /RHO/RHORH(100, 2), RHOVR(100, 2)
INTEGER BLDATA, AANDK, ERSOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
1 UPPER, St, St, SRF
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIM1
C
C READ AND PRINT ALL INPUT DATA
WRITE(6,1000)
READ(5,1100)
WRITE(6,1100)
READ(5,1030) GAM,AR,TIP,RHOT,WFL,BLANK,OMEGA,ORF
WRITE(6,1040) GAM,AR,TIP,RHOT,WFL,BLANK,OMEGA,ORF
WRITE(6,1120)
READ(5,1030) BETAI,BETA0,CHORD(I),STGR(I)
WRITE(6,1040) BETAI,BETA0,CHORD(I),STGR(I)
WRITE(6,1130)
READ(5,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
WRITE(6,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
DO 10 J = 1,2
IF (J.EQ.1) WRITE(6,1140)
IF (J.EQ.2) WRITE(6,1150)
WRITE(6,1180) J,J,J,J,J,
READ(5,1030) K(J),KO(J),BETI(J),BET0(J),SPLNO
WRITE(6,1040) K(J),KO(J),BETI(J),BET0(J),SPLNO
NSPI(J) = SPLNO
NSP = NSPI(J)
WRITE(6,1190) J,
READ(5,1030) {MSP(I,J),I=1,NSP}
WRITE(6,1040) {MSP(I,J),I=1,NSP}
WRITE(6,1200) J,
READ(5,1030) {THSP(I,J),I=1,NSP}
10 WRITE(6,1040) {THSP(I,J),I=1,NSP}
WRITE(6,1210)
READ(5,1030) {MR(I),I=1,NRSPI}
WRITE(6,1040) {MR(I),I=1,NRSPI}
WRITE(6,1220)
READ(5,1030) {AMSP(I),I=1,NRSP}
WRITE(6,1040) {AMSP(I),I=1,NRSP}
WRITE(6,1230)
READ(5,1030) {BESP(I),I=1,NRSP}
WRITE(6,1040) {BESP(I),I=1,NRSP}
WRITE(6,1240)
READ(5,1010) BLDAT,ANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
WRITE(6,1020) BLDAT,ANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
IF (MM.LT.100.AND.MM.LT.50 .AND. NRSPI.LT.50 .AND. NSPI(1).LT.50
1 .AND. NSPI(2).LT.50) GO TO 20
WRITE(6,1250)
STOP

C CALCULATE MV ARRAY

C 20 HM1 = CHORD(I)/FLOAT(MBO-MBI)
DO 30 IM=1,MM
30 MV(IM) = FLOAT(IM-MBI)*HM1

C CALCULATE MISCELLANEOUS CONSTANTS

C NER(1)=0
NER(2)=0
PITCH = 2*3.1415927/FLOAT(NBL)
HT= PITCH/FLOAT(NBL)
DMLR= HT/1000.
DMLR= HM1/1000.
BV(1) = 0.
BV(2) = 1.
MBIM1= MB1-1
MBIP1= MB1+1
MBGM1= MBO-1
MBOP1= MBO+1
MM1 = MM-1
CP = AR/(GAM-1.)*GAM
EXPON= 1./(GAM-1.)
TWW= 2.*OMEGA/WTFL
CPT= 2.*CP*Trp
TGRUG= 2.*GAM*A~/IGAM-1.)
CALL SPLINT(MR,RMSP,NRSP,MV,MM,RM,SAL)
CALL SPLINT(MR,BESP,NRSP,MV,MM,BE,OBDM)

C CALCULATE GEOMETRICAL CONSTANTS

CHORD(I) = CHORD(I)
STGR(I) = STGR(I)
MLE(I) = 0.
MLE(2) = 0.
THLE(I) = 0.
THLE(2) = PITCH
RM(1) = RM(MBI)
RM(2) = RM(MBI)
RMU(1) = RM(MBI)
RMU(2) = RM(MBI)

C INITIALIZE ARRAYS

DO 60 I=1,2500
  UI(I) = 1.
  K(I) = 0.
60 RHO(I) = RHOIP

DO 70 IM=1,100
  DO 70 SURF=1,2
    RHOH8(IM,SURF) = RHOIP
70 RHOV8(IM,SURF) = RHOIP
RETURN

1000 FORMAT (1H1)
1010 FORMAT (16(5))
1020 FORMAT (1X,16I7)
1030 FORMAT (8F10.5)
1040 FORMAT (1X,8G16.7)
1100 FORMAT (I10)
1110 FORMAT (7X,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRH0IP,12X,4HWTL,11X,6HW
        LTFLSP,10X,5HOMEGA,12X,3HORF)
1120 FORMAT (6X,5HBE,10X,5HBETAO,11X,6HCHORDF,11X,5HSTGRF)
1130 FORMAT (41H MB, MBI, MBO, NBI, NBL, NRSP)
1140 FORMAT (39HL BLADE SURFACE 1 -- UPPER SURFACE)
1150 FORMAT (39HL BLADE SURFACE 2 -- LOWER SURFACE)
1160 FORMAT (7X,2HRI,11,12X,2HRG,11,12X,4HBETI,11,11X,4HBETO,11,11X,5HS
        IPLNO,11)
1190 FORMAT (7X,3HMS,11,2X,5HARRAY)
1200 FORMAT (7X,4HTSHP,11,2X,SHARRAY)
1210 FORMAT (16HL MR ARRAY)
1220 FORMAT (7X,11HRMSP ARRAY)
1230 FORMAT (7X,11HBESP ARRAY)
1240 FORMAT (52HL BLDAT AANDK ERSOR STRFN SLCRD INTVL SURVL)
1250 FORMAT (41H1 MM,NBHI,NRSP,OR SOME SPLNO IS TOO LARGE)
END
SUBROUTINE PRECAL

PRECAL CALCULATES ALL REQUIRED FIXED CONSTANTS

COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON /AKRU/ AH(2500), AL(2500), K(2500), RH0(2500)
COMMON /INP/GAM, AR, TIP, RH0IP, WFL, OMEGA, ORF, BETAI, BETAO,
1 MBi, MBO, MM, NBBI, NBL, NRSP, MR(50), RMSP(50), BESP(50),
2 BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/ MBIMI, MBIP, MBOMI, MBOM, MMH, HT, DILX, DMLR,
1 PITCH, CP, EXPON, TWL, CPTIP, TGROG, TBO, LAMDA, TWL, ITMIN, ITMAX,
2 NIP, INS(2), V(2), MV(100), TV(100), TV(100, 2),
3 DTDW(100, 2), BETAV(100, 2), MH(100, 2), DTMH(100, 2), BETAH(100, 2),
4 RMH(100, 2), BEH(100, 2), RMSP(100), DTMSP(100), SAL(100),
5 AAA(100)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
1 UPPI, SI, ST, SRW
REAL K, KAK, LAMDA, LMAX, MF, MLF, MR, MSL, MSP, MV, MVIMI
EXTERNAL BL1, BL2

C CALCULATE LAMBDA AND VI
C
BETAI = BETAI/57.295779
BETAO = BETAO/57.295779
TBI = SIN(BETAI)/COS(BETAI)
TBO = SIN(BETAO)/COS(BETAO)
10 RHOT = RH0IP
RHOVI = WFL/BE(MBI)/PITCH/COS(BETAI)/RM(MBI)
20 VI = RHOVI/RHOT
LAMDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))
TTIP = 1.-((VI**2+2.*OMEGA*LAMBOA-(OMEGA*RMIMBII)**2)/CPTIP
IF(TTIP.LE.0.) GO TO 30
RMHMBI = RHOIOPT*TTIP**EXPON
IF(ABS(RMHBMI-RHOT)/RHOT.LT..000001) GO TO 40
RHOT = RHOMBI
GO TO 20
30 WFL = WFL/2.
NER(2) = NER(2)+1
WRITE(5,10200) WFL
IF(NER(2).EQ.10) STOP
GO TO 10
40 VI = RH0V/RH0MBI
LAMDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))

C CALCULATE MAXIMUM VALUES FOR RHO*W AT LEADING AND TRAILING EDGE

TWL = 2.*OMEGA*LAMDA
AA = (TWL-(OMEGA*RM(MBI))**2)/CPTIP
TPP = TIP*(1.-AA)
BB = TGROG*TPP
TTIP = 1.-BB/CPTIP-AA
RHOT = RHOIP*TTIP**EXPON
RHOMBI = RHOT*SQRT(BB)
AA = (TWL-(OMEGA*RM(MBI))**2)/CPTIP
TPP = TIP*(1.-AA)
BB = TGROG*TPP
TTIP = 1.-BB/CPTIP-AA
RHOT = RHOIP*TTIP**EXPON
RHOMBO = RHOT*SQRT(BB)

CALCULATE V0 AND W-CRITICAL AT BLADE LEADING AND TRAILING EDGE
RHVO = WTFL/PITCH/COS(BETA0)/RM(MBO)
RHOMB2 = RHOIP
TWLMR = TWL-(OMEGA*RM(MBO))**2
LER(I) = 1
CALL DENSITY CALL NO. 1
CALL DENSITY(RHVO, RHOMB2, VO, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
WCRI = SQRT((TGR0G*TIP*(1.-TWL-(OMEGA*RM(MBO))**2)/CPTIP))
WCRO = SQRT((TGR0G*TIP*(1.-TWL-(OMEGA*RM(MBO))**2)/CPTIP))

C CALCULATE BETA CORRECTED TO BOUNDARY A-N AND G-H
C
TWLMR = TWL-(OMEGA*RM(MBII)**2
RHOM = RHOMB2
TBII = 1.E20
50 TBII = (TBII/BMI)*RHO1/RHOMBI+OMEGA*(RM(MBI)**2-RM1**2)*RHO1
1 /WTFL*PITCH*BE(1)
IF(ABS(TBII-TBIT).LT.0.00001) GO TO 60
TBII = TBIT
RHOVI = WTFL/PITCH*SQRT(1.+TBII**2)/BE(1)/RM(1)
LER(I) = 2
CALL DENSITY CALL NO. 2
CALL DENSITY(RH0VI, RHR01, AA, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
GO TO 60
60 TBIT = TBII
BTAIN = ATAN(TBII)*57.295779
TWLMR = TWL-(OMEGA*RM(MM))**2
RHOMM = RHOMB2
TBOM = 1.E20
70 TBOM = (TBOM/BEM)*RHO/M/RHOM/RHOMB2+OMEGA*(RM(MBO)**2-RM(MM)**2)*1 /WTFL/PITCH*BE(MM)
1
IF(ABS(TBOM-TBOT).LT.0.00001) GO TO 80
TBOM = TBOT
RHOO = WTFL/PITCH*SQRT(1.+TBDM**2)/BE(MM)/RM(MM)
LER(I) = 3
CALL DENSITY CALL NO. 3
CALL DENSITY(RHOO, RHOMH, AA, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
GO TO 70
80 TBO = TBOT
BTOUT = ATAN(TB0)*57.295779
C CALCULATE TV, ITV, IV, DTDMV, AND BETAV ARRAYS
C
, ITMIN = 0
ITMAX = NBBI-1
C TV, ITV, AND DTDMV ON BLADE
DO 90 IM=MBI,MBO
LER(2) = 1
CALL BLCD CALL NO. 1
CALL BL1(MVM(IM), TV(IM,1), DTDMV(IM,1), INF)
ITV(IM,1) = INT((TV(IM,1)-DTLR)/HT)
IF (TV(IM,1)-DTLR) ITV(IM,1) = ITV(IM,1)+1
ITMIN = MINO(ITMIN, ITV(IM,1))
LER(2) = 2
CALL BLCD CALL NO. 2
CALL BL2(MVM(IM), TV(IM,2), DTDMV(IM,2), INF)
ITV(IM,2) = INT((TV(IM,2)-DTLR)/HT)
IF (TV(IM,2)-DTLR) ITV(IM,2) = ITV(IM,2)+1
90 ITMAX = MAXO(ITMAX, ITV(IM,2))
C ITV AND IV UPSTREAM OF BLADE
FIRST = 0
LAST = NBBI-1
DO 120 IM=1,MBIM1
ITV(IM,1) = FIRST
120 ITV(IM,2) = LAST
C ITV DOWNSTREAM OF BLADE

27
LAST = ITV(MBO, 2)
FIRST = LAST + 1 - NVBI
DO 150 IM = MBOPL, MM
ITV(IM, 1) = FIRST
150 ITV(IM, 2) = LAST
ITMIN = MINO(ITMIN, ITV(MM, 1))

CALCULATE IV ARRAY
IV(1) = 1
DO 160 IM = 1, MM
160 IV(IM + 1) = IV(IM) + ITV(IM, 2) - ITV(IM, 1) + 1

BETAV ARRAY
DO 200 SURF = 1, 2
DO 200 IM = MBI, MBO
200 BETAV(IM, SURF) = ATAN(DTDMV(IM, SURF)) * RM(IM) * 57.29579
NIP = IV(MM) + NVBI - 1
WRITE(6, 1030) VI, RHOWMI, WCRI, BTAIN, VO, RHOWKO, WCRO, GTAOUT
WRITE(5, 1050) IM, IV(MM), ITMAX, LAMBDALNIP
WRITE(6, 1040) PI, PIH, HMI

IF (BLDAT .LE. 0) GO TO 230
WRITE (6, 1080) IM, TV(IM, 1), DTDMV(IM, 1), TV(IM, 2), DTDMV(IM, 2),
          TV(IM, 2), RM(IM, 1), SAL(IM), BE(IM), DBDM(IM), IM = 1, MM
230 CONTINUE

C CALCULATE MH AND DTDMH ARRAYS
C
ITO = ITV(1, 1)
MRTS = 1
IMS(1) = 1
MH(1, 1) = 0.
DTDMH(1, 1) = 1.E10
LER(2) = 3
C
BLCD AND ROOT (VIA MHORIZ) CALL NO. 3
CALL MHORIZ(MV, ITV(1, 1), BL1, MBI, MBO, ITO, HT, DTLR, O, IMS(1), MH(1, 1),
          1 DTDMH(1, 1), MRTS)
IF (ITV(MBO, 1) - ITV(MBO, 2) + NVBI .NE. 2) GO TO 240
IMSL = IMS(1) + 1
MH(IMSL, 1) = MV(MBO)
DTDMH(IMSL, 1) = -1.E10
IMS(1) = IMSL
240 IMS(2) = 0
MRTS = 1
LER(2) = 4
C
BLCD AND ROOT (VIA MHORIZ) CALL NO. 4
CALL MHORIZ(MV, ITV(1, 2), BL2, MBI, MBO, ITO, HT, DTLR, O, IMS(2), MH(1, 2),
          1 DTDMH(1, 2), MRTS)
I = MAX0(IMS(1), IMS(2))
IF (I .LE. 100) GO TO 290
WRITE(6, 1100) I
STOP
290 IF (BLDAT .LE. 0) GO TO 300
WRITE (6, 1110) (IM, IV(IM), (ITV(IM, SURF), SURF = 1, 2), IM = 1, MM)
C
C CALCULATE RMH, BEH, AND BETAH ARRAYS
C
300 IF (BLDAT .GT. 0) WRITE(6, 1120)
    DO 320 SURF = 1, 2
        CALL SPLINT(MR, RMS, NRSP, MH(1, SURF), IMS(SURF), RMH(1, SURF), AAA)
        CALL SPLINT(MR, BE, NRSP, MH(1, SURF), IMS(SURF), BEH(1, SURF), AAA)
        IMS(SURF) = IMS(SURF)
        IF (IMSS .LT. 1) GO TO 320
    DO 310 IMS = 1, IMSS
310 BETAH(IHS,SURF) = ATAN(DTDMH(IHS,SURF)*RMH(IHS,SURF))*57.295779
IF (BLDAT.GT.0) WRITE(6,1130) SURF,LMH(IM,SURF),RMH(IM,SURF),
 1 BETAH(IM,SURF),BETAH(IM,SURF),DTOMH(IM,SURF),IM=1,IMSS)
320 CONTINUE
IF (BLDAT.LE.0) GO TO 340
WRITE (6,1140)
IT = ITMIN
330 IF (IT.GT.ITMAX) GO TO 340
TH = FLOAT(IT)*HT
WRITE (6,1010) IT,TH
IT = IT+1
GO TO 330
340 IF(NIP.LE.2500) GO TO 350
WRITE(6,L150)
STOP
350 WRITE (6,1000)
RETURN
1000 FORMAT (1H1)
1010 FORMAT (4X,I4,G16.5)
1020 FORMAT(60H INPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE LEADING
 1 EDGE/6H WTFL REDUCED TO,G14.6)
1030 FORMAT (1H1/24X,10HFREESTREAM,8X,13HMAXIMUM VALUE,
 17X,6HCRITICAL,3OX,14HBEATA CORRECTED/25X,8HVELOCITY,10X,9HFOR RHO*W
 2,10X,9HVELOCITY,31X,11HTO BOUNDARY/1X,17HLEADING EDGE B-G,3G18.5,
 312X,12HBOUNDARY A-H,3GB18.5,11X,17HTRAILING EDGE C-F,3G18.5,12X,
 142HBOUNDARY D-E,3G18.5)
1040 FORMAT(33Hl CALCULATED PROGRAM CONSTANTS//5X,5HPITCH,13X,
 1 2HHT,13X,3HHMI/1X,5G16.7)
1050 FORMAT (15X,5HITMIN,10X,5HITMAX/4X,15,10X,5HITMAX/3X,6HLAMRDA/1X,G16.7)
 1 ,/38HL NUMBER OF INTERIOR MESH POINTS = ,15)
1060 FORMAT(33HL SURFACE BOUNDARY VALUES//5X,7HSURFACE,1X,2HBV
 1/(5X,14,4X,F10.5))
1070 FORMAT (1HL,6X,62HBLADE DATA AT INTERSECTIONS OF VERTICAL MESH LIN
 1 ES WITH BLADES)
1080 FORMAT (1HL,22X,15HBLADE SURFACE,1,15X,15HBLADE SURFACE 2/7X,
 1 1LM,14X,2HTV,11X,5HDOTTW,12X,2HTV,11X,5HDOTTW/(5G15.5)
1090 FORMAT (1HL,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABLE /1
 1 2X,2HIM,7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HVB/DM/11X,13,
 2 5G15.5))
1100 FORMAT(34Hl ONE OF THE MH ARRAYS IS TOO LARGE/7H IT HAS,15, 8H POI
 1 NTS)
1110 FORMAT (4H1 IM,9X,8HTV ARRAY,25X,9HTV ARRAY/38X,5HBLADE/37X,7HSUR
 1 FACE,3X,1HL,5X,1H2/39X,3HNO./11X,13,5X,110,25X,2(14,2X))
1120 FORMAT (67HIM COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LIN
 1 E WITH BLADE)
1130 FORMAT (25HLMH ARRAY - BLADE SURFACE,12,7X,2HMMH,19X,3HRMH,19X,
 1 3HBEC,18X,5HBECATIES,17X,5HDOTTW/(5G22.4))
1140 FORMAT (43HTHETA COORDINATES OF HORIZONTAL MESH LINES/6X,2HT,
 15X,5HBETA)
1150 FORMAT(48Hl THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500)
END
SUBROUTINE MHORIZ(MV,ITV,MBI,MBO,ITO,HT,DTLR,KODE,J,MH,DTDMH,IMRTS)

C MHORIZ CALCULATES M COORDINATES OF INTERSECTIONS OF ALL HORIZONTAL MESH LINES WITH A BLADE SURFACE
C KODE = 0 FOR UPPER BLADE SURFACE
C KODE = 1 FOR LOWER BLADE SURFACE
C
COMMON SRW,ITER,IERD,LER(2),NER(2)
DIMENSION MV(IOO),ITV(IOO),MH(IOO),DTDMH(IOO)
INTEGER BLDATA,AANDK,ERSOR,STRFY,SLCRD,SURVL,AATEMP,SURF,FIRST,
UPPER,SI,ST,SRW
REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM
REAL MVIM
EXTERNAL BL
IF (MBI.GE.MBO) RETURN
IM= MBI
10 ITIND= 0
20 IF (ITV(IM+1)-ITV(IM)-ITIND) 30,40,50
30 J= J+1
   TI= FLOAT(ITV(IM+1)-ITO-ITIND*KODE)*HT
   ITIND= ITIND-1
   MVIM = MV(IM)
   IF (MRRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
   CALL ROOT (MVIM,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
   GO TO 20
40 IM= IM+1
   MRTS = 0
   IF (IM.EQ.MBO) RETURN
   GO TO 10
50 J= J+1
   TI= FLOAT(ITV(IM)-ITO+ITIND*KODE)*HT
   ITIND= ITIND+1
   MVIM = MV(IM)
   IF (MRRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
   CALL ROOTT(MVIM,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
   GO TO 20
END
SUBROUTINE COEF

COEF CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K, 
AT ALL UNKNOWN MESH POINTS FOR THE ENTIRE REGION

COMMON SRW,ITER, IEND, LER(2), NER(2)
COMMON /AUKRHO/ A(2500, 4), U(2500), K(2500), RHO(2500)
COMMON /INP/GAM, AR, TIP, RHOIP, WTFL, OMEGA, ORF, BETAI, BETAO,
  1 MBI, MBO, MM, NBBI, NBL, NSP, MR(50), RMS(50), BESP(50),
  2 BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/MBIM1, MBIPI, MBOPI, MBI, MBO, MM, NBBI, NBL,
  1 MR, MBI, MBO, MM, NBBI, NBL, NSP, SRP(50), BESP(50),
  2 BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /HRBAAK/H(4), R(4), B(4), KAK(4), KA(4), RZ, BZ, IH(4)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
  1 UPPER, SI, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIM1

C INITIALIZE ARRAYS
  ITEK = ITER+1
  IH(1) = 1
  IH(2) = 0

C INCOMPRESSIBLE CASE
  IF(GAM.NE.1.5 OR AR.NE.1000 OR TIP.NE.1.E6) GO TO 20
   IEND = 1
   GO TO 40

C ADJUSTMENT OF PRINTING CONTROL VARIABLES
  IF(ITER.NE.1 AND ITER.NE.2) GO TO 30
    AANDK = AANDK-1
    ERSOR = ERSOR-1
    STRFN = STRFN-1
    SLCRD = SLCRD-1
    INTVL = INTVL-1
    SURVL = SURVL-1
  30 IF(IEND.NE.0) GO TO 40
    AANDK = AANDK+2
    ERSOR = ERSOR+2
    STRFN = STRFN+2
    SLCRD = SLCRD+2
    INTVL = INTVL+2
    SURVL = SURVL+2

C FIRST VERTICAL MESH LINE
  DO 50 IP=1, NBBI
    A(IP,1) = 0.
    A(IP,2) = 0.
    A(IP,3) = 0.
  50
A(IP,4) = 1.
50 K(IP) = HML*IBI/PITCH/RM(1)
C
C UPSTREAM OF BLADE, EXCEPT FOR FIRST VERTICAL MESH LINE
C
IF(2.GT.MBIM1) GO TO 70
DO 60 IM=2,MBIM1
60 CALL CDEFP(IM)
C
C BETWEEN BLADES
C
70 DO 80 IM=MBI,MBO
80 CALL COEFBB(IM)
C
C DOWNSTREAM OF BLADES EXCEPT FOR FINAL MESH LINE
C
150 IF(MBOP1.GT.MM1) GO TO 170
DO 160 IM=MBOP1,MM1
160 CALL COEFPI(IM)
C
C FINAL VERTICAL MESH LINE
C
170 IVMM = IV(MM)
   DO 180 IP=IVMM,MM1
       A(IP,1) = 0.
       A(IP,2) = 0.
       A(IP,3) = 1.
       A(IP,4) = 0.
180 K(IP) = -HML*IBI/PITCH/RM(MM)
C
C TAKE CARE OF POINTS ADJACENT TO B, AND CASES WHEN POINTS J,C,E, OR F
C ARE GRID POINTS
C
POINT B
   IP = IV(MBIM1)
   A(IP,4) = 0.
C
POINT C
   IF(IV(MBO,1)-IV(MBO,2)+NBBI.NE.2) RETURN
   IT = IV(MBO,1)-1
   IP = IPF(MBOP1,IT)
   A(IP,3) = 0.
RETURN
END
$IBFTC COEFBB DEBUG

SUBROUTINE COEFBB(IM)
C COEFBB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K ALONG ALL VERTICAL MESH LINES WHICH INTERSECT BLADES
C
COMM A/AUKRHO/ A(2500,4), U(2500), K(2500), RH0(2500)
COMM /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,OMF,BETA1,BETA2,
  1 MB1,MBO,MNBBI,NBL,NRSP,MR(50),RMRSP(50), BESP(50),
  2 BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
COMM /CALCON/MBIM,MBIP,MBOM,MBOP,MBM1,HMI,HT,DTLR,OML,
  1 PITCH,CP,EXPO,TW,TIP,TGROG,TBI,BBO,LAMDA,THL,ITMIN,ITMAX,
  2 NIP, IMS(2), BV(2), MVI(100), IV(100), IV(100,2), TV(100,2),
  3 DTMV(100,2), BEH(V(100,2), MH(100,2), DTMH(100,2), BETAH(100,2),
  4 RMH(100,2), BEOH(100,2), RM(100), BEOH(100), DBDM(100), SAI(100),
  5 AAA(100)
COMM /HRBAAK/H(4), R(4), B(4), KAK(4), KA(4), RZ, BZ, IH(4)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
  1 UPPER,Sl,ST,SRW
REAL K,KAK,LAMDA,LMAX,MH,MLE,MRL,MSG,MS,MV,MV1M1
IF (ITV(IM,1).GT. ITV(IM,2)) RETURN
ITVU = ITV(IM,1)
IT = ITV - 1
ITVL = ITV(IM,2)
IPU = IPT(IM,ITVU)
IPL = IPU+ITVL-ITVU
DO 90 IP=IPU,IPL
  IT = IT+1
  CALL HRB(IM,IT,IP)
  KAK(I) = O.
  DO 90 IF(IPA().EQ.I) A(IP,1) = O.
RETURN
C FIX HRB VALUES FOR CASES WHERE MESH LINES INTERSECT BLADES
60 IF(IT.EQ. ITV(IM,1)) CALL BDRY12(1,IM,IT)
IF(IT.EQ. ITV(IM,2)) CALL BDRY12(2,IM,IT)
ITVM1 = ITV(IM-1,1)
ITVP1 = ITV(IM-1,1)
IF(IT.LT. ITVM1) CALL BDRY34(3,IM,1)
IF(IT.LT. ITVP1) CALL BDRY34(4,IM,1)
IF(IT.GT. ITV(IM-1,2)) CALL BDRY34(3,IM,2)
IF(IT.GT. ITV(IM-1,2)) CALL BDRY34(4,IM,2)
70 IF(IM.EQ.MBO.AND.LOWER.EQ.2) GO TO 80
C COMPUTE A AND K COEFFICIENTS
80 CALL AAK(IM,IP)
  DO 90 I=1,4
    K(IP) = K(IP)*KAK(I)*A(IP,1)
  90 IF(KA(I).EQ.1) A(IP,1) = O.
RETURN
C C COEFP CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K ALONG ALL VERTICAL MESH LINES WHICH DO NOT INTERSECT BLADES
ENTRY COEFP(IM)
ITVU = ITV(IM,1)
IT = ITVU-1
ITVL = ITV(IM,2)
IPL = ITV(IM+1)-1
IPU = ITV(IM)
DO 100 IP=IPU,1PL
   IT = IT+1
   CALL HB(IM,IT,IP)
   IF (IT.EQ. ITVU) R(1) = RHO(IP)
   IF (IT.EQ. ITVL) R(2) = RHO(IPU)
100  CALL AAK(IM,IP)
      K(IPL) = K(IPL)+A(IPL,2)
      K(IPU) = K(IPU)-A(IPU,1)
END

$IBFC HRH DEBUG

SUBROUTINE HRB(IM,IT,IP)
C
C HRB CALCULATES MESH SPACING, H, DENSITIES, RZ AND R, AT GIVEN AND
C ADJACENT POINTS, AND STREAM SHEET THICKNESSES, BZ AND R, AT GIVEN
C AND ADJACENT POINTS
C
COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
COMMON /CALCON/MB(IM1),MB1P1,MBQMI,MBQPI,MMMI,H,M,L,TW,DLR,DLR,
1  PITCH,MCP,EXPON,TW,TPI,TP,TRB,LAMBDA,TWL,ITMIN,ITMAX,
2  NIP,IM(2),IV(2),MV(100),LV(101),ITV(100,2),TV(100,2),
3  TDM(100,2),BETAV(100,2),MH(100,2),OM(100,2),BETAM(100,2),
4  RMH(100,2),BEH(100,2),RM(100),BE(100),OM(100),SAL(100),
5  AAA(100)
COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),K(4),RZ,BZ,IP(4)
INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRO,SRVL,AATEMP,SURF,FIRST,
1  UPPER,SL,ST,SMW
REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
H(1) = HT*M(1M)
H(2) = HT*M(1M)
H(3) = MV(IM)-MV(IM-1)
H(4) = MV(IM+1)-MV(IM)
RZ = RHO(IP)
IP3 = IPF(IM-1,IT)
IP4 = IPF(IM+1,IT)
R(1) = RHO(IP-1)
R(2) = RHO(IP+1)
R(3) = RHO(IP3)
R(4) = RHO(IP4)
BZ = BE(IM)
B(31 = BE(IM-1)
B(4) = BE(IM+1)
RETURN
END
SUBROUTINE AAK(IP, IM)
C
AAK Calculates finite difference coefficients, A, and constant, K,
C at a single mesh point
C
COMMON /AUKRHO/, A(2500, 4), U(2500), K(2500), RHO(2500)
COMMON /CALCON/ MBIM1, MBIP1, MBDM1, MBPO1, MMML, HH1, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, TW, CPTIP, TROG, TBI, TBO, LAMDA, TWL, ITMIN, ITMAX,
2 NIP, IMS(2), BV(2), MV(100), IV(101), TV(100, 2), TV(100, 2),
3 DTDV(100, 2), BETAV(100, 2), MH(100, 2), DTDH(100, 2), BETAH(100, 2),
4 RMH(100, 2), BEH(100, 2), RM(100), BE(100), DBDM(100), SAL(100),
5 AAA(100)
COMMON /HRBAAK/H(4), R(4), B(4), KA(4), RZ, BZ, IH(4)
INTEGER BLDAT, AANDK, SR, STRFN, SLCRD, SURV, AATEMP, SURF, FIRST,
1 UPPER, SI, ST, SRW
REAL K, KAK, LAMBDAt, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIML
A12 = Z./H(I)/H(2)
A34 = 2./H(3)/H(4)
AZ = A12 + A34
B12 = (I/2) - R(I+1) / RZ / (H(I)+H(2))
B34 = (B(I+4)*R(I+4) - B(I+3)*R(I+3)) / BZ / RZ / (H(3)+H(4)) - SAL(IM) / RM(IM)
A(IP, 1) = (Z./H(I)+B12) / AZ / (H(I)+H(2))
A(IP, 2) = A12 / AZ - A(IP, 1)
A(IP, 3) = (Z./H(I)+B34) / AZ / (H(3)+H(4))
A(IP, 4) = A34 / AZ - A(IP, 3)
K(IP) = -TWW * BZ * RZ * SAL(IM) / AZ
RETURN
END

SUBROUTINE BDRY12(IP, IM, IT)
C
BDRY12 corrects values computed by HRB when a vertical mesh line
C intersects a blade
C
COMMON /CALCON/ MBIM1, MBIP1, MBDM1, MBPO1, MMML, HH1, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, TW, CPTIP, TROG, TBI, TBO, LAMDA, TWL, ITMIN, ITMAX,
2 NIP, IMS(2), BV(2), MV(100), IV(101), TV(100, 2), TV(100, 2),
3 DTDV(100, 2), BETAV(100, 2), MH(100, 2), DTDH(100, 2), BETAH(100, 2),
4 RMH(100, 2), BEH(100, 2), RM(100), BE(100), DBDM(100), SAL(100),
5 AAA(100)
COMMON /RHOOHRH/H(4), R(4), B(4), KA(4), RZ, BZ, IH(4)
INTEGER BLDAT, AANDK, SERSOR, STRFY, SLCRD, SURV, AATEMP, SURF, FIRST,
1 UPPER, SI, ST, SRW
REAL K, KAK, LAMDBA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIML
H(I) = ABS FLOAT(I+1) * HT - TV(IM, I) * RM(IM)
R(I) = RHOVBI(1M, I)
KAK(1) = BV(I)
KAAK = 1
RETURN
END
SUBROUTINE BDRY34(I, IM, SURF)

C BDRY34 CORRECTS VALUES COMPUTED BY HRB WHEN A HORIZONTAL MESH LINE
C INTERSECTS A BLADE

COMMON /CALCON/MBIM1,MBIPL,MBOL1,MMMI,MMH1,HT,DILR,DMLR,
1 PITCH,CP,EXPN,TW,TFTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
2 NIP,INS(2),BVI(2),MVI(100),TVI(101),TVI(100,2),TVI(100,2),
3 DTMVI(100,2),BETA(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
4 RMH(100,2),BEMH(100,2),RM(100),BEM(100),DBDM(100), SAL(100),
5 AAA(100)

COMMON /RHO/RHOB(100,2),RHOVB(100,2)

COMMON /HRBAAK/H4),R4),B4),K(4),K(4),RZ,BZ,IH(4)

INTEGER BLDAT,AAANDK,ERSOR,STRFN,SLCRD,SURVL,AAATEMP,SURF,FIRST,
1 UPPER,SI,ST,SRW

REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1

IH(ISURF)=IH(ISURF)+1

IH(SURF)=AB(MV(1M)-MH(IHS,SURF))

R(I)=RHOHB(IHS,SURF)

B(I)=BEH(IHS,SURF)

KAK(I)=BV(ISURF)

KAI(I)=1

RETURN

END

SUBROUTINE SOR

C SOR SOLVES THE SET OF SIMULTANEOUS EQUATIONS FOR THE STREAM FUNCTION
C USING THE METHOD OF SUCCESSIVE OVER-RELAXATION

COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)

COMMON /INP/GAM,AR,TIP,RHOIP,MTFL,OMEGA,ORF,BETAI,BETA0,
1 MB1,MB2,MM,MB81,MB11,NBBI,NBL,NRSP,MR(50),RMS(50),BESP(50),
2 BLDAT,AAANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL

COMMON /CALCON/MBIM1,MBIPL,MBOL1,MMMI,MMH1,HT,DILR,DMLR,
1 PITCH,CP,EXPN,TW,TFTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
2 NIP,INS(2),BVI(2),MVI(100),TVI(101),TVI(100,2),TVI(100,2),
3 DTMVI(100,2),BETA(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
4 RMH(100,2),BEMH(100,2),RM(100),BEM(100),DBDM(100), SAL(100),
5 AAA(100)

INTEGER BLDAT,AAANDK,ERSOR,STRFN,SLCRD,SURVL,AAATEMP,SURF,FIRST,
1 UPPER,SI,ST,SRW

REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1

AAATEMP = AAANDK

IF (ORF.GE.2.) ORF=0.

IF(ORF.GT.1.) GO TO 50

ORF = 1.

ORF= 2.

40 ORFTEM=ORF

LMAX = 0.

50 IF(AAATEMP.GT.0) WRITE(6,1010)

ERROR = 0.

C SOLVE MATRIX EQUATION BY SOR, OR CALCULATE OPTIMUM OVERRELAXATION
C FACTOR

C
IP = 0
DO 120 IM=1,MM
   IPU = IV(IM)
   IPL = IV(IM+1)-1
   ITVU = ITV(IM,1)
   IF(AATEMP.GT.0) WRITE (6,1020) IM,IT
   DO 120 IP=IPU,IPL
      IP1 = IP-1
      IP2 = IP+1
      C CORRECT IP1 AND IP2 ALONG PERIODIC BOUNDARIES
      IF(IP.GE.MBI.AND.IP.LE.MBO) GO TO 60
      IF(IT.EQ.ITV(IM,1)) IP1 = IP1+VBBI
      IF(IT.EQ.ITV(IM,2)) IP2 = IP2-NBBI
   60 IT3 = IT
   IT4 = IT
   100 IP3 = IPF(IM-1,IT3)
   IP4 = IPF(IM+1,IT4)
   IF(ORF.GT.1.) GO TO 110
   C CALCULATE NEW ESTIMATE FOR LMAX
   UNEW = U(IP,1)*U(IP)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4)
   IF(UNEW.LT.1.E-25) U(IP) = 0.
   IF(U(IP).EQ.0.) GO TO 115
      RATIU = UNEW/U(IP)
      LMAX = AMAX1(RATIO,LMAX)
      U(IP) = UNEW
   GO TO 115
   110 CHANGE = ORF*(K(IP)-U(IP)+A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4))
   ERROR = AMAX1(ERROR,ABS(CHANGE))
   U(IP) = U(IP)+CHANGE
   115 IF(AATEMP.LE.0) GO TO 120
   WRITE (6,1030) IT,IP,IP1,IP2,IP3,IP4,A(IP,1),A(IP,2),A(IP,3),A(IP,4)
120 IT = IT+1
   AATEMP = 0
   IF(ORF.GT.1.) GO TO 130
   ORFOPT = 2/((1.+SQRT(ABS(1.-LMAX))))
   WRITE(6,1000) ORFOPT
   IF(ORFTEM-ORFOPT.GT.0.00001.OR.ORFOPT.GT.1.999) GO TO 40
   WRITE (6,1070)
   ORF = ORFOPT
   GO TO 50
130 IF(ERSUR.GT.0) WRITE(6,1040) ERROR
   IF(ERSUR.GT.0.00001) GO TO 50
   IF(STRFN.LE.0) RETURN
   C PRINT STREAM FUNCTION VALUES FOR THIS ITERATION
   C WRITE (6,1050)
   DO 140 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      ITVU = ITV(IM,1)
      WRITE (6,1020) IM,ITVU
   140 WRITE (6,1060) (U(IP),IP=IPU,IPL)
   RETURN
1000 FORMAT(6H ESTIMATED OPTIMUM ORF =,F9.6)
1010 FORMAT (6H IT [IP] IP1 IP2 IP3 IP4 A(1) A(2) A(3) A(4) K)
1020 FORMAT(6H ERROR =,F11.8)
1030 FORMAT(6H STREAM FUNCTION VALUES)
1040 FORMAT (6H RETURN)
END
SUBROUTINE SLAX

SLAX CALLS SUBROUTINES TO CALCULATE RHO*W-SUB-M THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND TO CALCULATE AND PLOT THE STREAMLINE LOCATIONS.

COMMON /AUKRHO/A(2500,4),U(2500),K(2500),RHO(2500)
COMMON /INP/GAM,AR,IP,RH0IP,WTL,OMEGA,DRF,BETAI,BETA0,
1 MB1,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BES(50),
2 BLDAT,ANMK,ERSOR,STRF1N,SLCRD,INTVL,SURVL
COMMON /CALCON/MBIM1,MBOPl,MBOMl,MMML,HMI,HT,DLR,OMLR,
1 PITCH,CP,EXPON,TW,CPTIP,TG406,TBI,TBO,LAMBD1A,W11,ITMIN,ITMAX,
2 NIP,IMS(2),BV(2),MV(100),IW(100),ITV(100,2),ITV100,2),
3 DTOMV(100,2),RETA1V(100,2),MH(100,2),OMH(100,2),BETAH(100,2),
4 RMH(100,2),BEH100,2),RM(100),BE(100),OMBD(100),SAL(100),
5 AAA(100)
COMMON /SLA/TSL(600),UINT(6)
DIMENSION MSL(100),KKK(14),P(4)
DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
1 XDOWN(400),YACROS(400)
EQUIVALENCE (A(1,1),W(1)),A(1,2),RWM(1),(A1,3),BETA(1)),
1 (A1,4),WM(1)),A(201,4),WTB(111),A(1401,4),XDOWN(111),
2 (K(1),YACROS(11)
INTEGER BLDAT,ANMK,ERSOR,STRF1N,SLCRD,INTVL,SURVL,AATEMP,SURF,FIRST,
1 UPPER,SL,SRW
REAL K,KAK,LMABDA,LMAX,MM,MLE,MR,MSL,MSP,MV,MVIM1
DATA (KK1(J),J=4,14,2)/61H*1#/

CALL SLAVP AND SLAVBB THROUGHOUT THE REGION

ITVU = ITV(1,1)
ITVL = ITV(1,2)
10 CALL SLAVP(IM,ITVU,ITVL)
DO 10 IM = MB1,MBO
10 ITVU = ITVU,IM,ITVL
DO 20 IM = MB1,MBO
20 CALL SLAVBB(IM)
90 ITVU = ITV(MB0PI,1)
ITVL = ITV(MBOPl,2)
DO 100 IM = MBOPl,MM
100 CALL SLAVP(IM,ITVU,ITVL)

PLOT STREAMLINES

IF (SLCRD.LE.0) RETURN
DO 110 IM = M1,MM
110 MSL(IM) = MV(IM)
K(01) = 7
K(02) = 6
K(01) = MM
P(1) = 1.
P(3) = 0.
P(4) = 0.
WRITE(5,1000)
CALL PLOTMY(MSL,TSL,KKK,P)
WRITE(6,1010)
RETURN
1000 FORMAT (2HPT,50X,16HSTREAMLINE PLOTS)
1010 FORMAT (2HPL,40X,70HSTREAMLINES ARE PLUTTED WITH THETA ACROSS THE
IPAGE AND M DOWN THE PAGE)
END
SUBROUTINE SLAV

C SLAV CALCULATES $\rho*w$-SUB-M THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND CALCULATES THE STREAMLINE LOCATIONS

COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON /AUKRHO/ AI2500, BM2500, U(2500), K(2500), RHO(2500)
COMMON /INP/GAM, AR, TIP, RHOIP, MTFL, OMEGA, ORF, BETA, ETAU,
1 MB1, MB0, MM, NBBI, N2L, NRSR, MR(50), RMSR(50), BESP(50),
2 BLDAT, AANDK, ESOR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/ MBIM1, MBIP1, MBOM1, MBBD1, MMM1, HM1, HT, DTLR, OMLR,
1 PITCH, CP, EXPON, TWW, CPTIP, TGRG, TBI, TRO, LAMBDA, TWL, ITMIN, ITMAX,
2 NIP, MSL(2), BVL(2), MV(100), IV(101), IVT(100, 2), TV(100), 2,
3 DTDMH(100, 2), BETAH(100, 2), DTDMMH(100, 2), BETAH(100, 2),
4 RH(100), BM(100), RW(100, 2), BEH(100, 2), BM(100), DBDM(100), SAL(100),
5 AAA(100)
COMMON /SLA/TSL(600), UINT(6)
DIMENSION TSP(50), USP(50), DUDT(50), TINT(6)
DIMENSION W(2500), RW(2500), BETA(2500), WMB(100, 2), WTB(100, 2),
1 XDWN(400), YACROS(400)
EQUIVALENCE (A(I, 1), W(I)), (A(I, 2), RW(I)), (A(I, 3), BETA(I)),
1 (A(I, 4), WMB(I)), (A(201, 4), WTB(I)), (A(401, 4), XDWN(I)),
2 (K(I), YACROS(I))
INTEGER BLDAT, AANDK, ESOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
1 UPPER, SL, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVI1

C SLAVP CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT INTERSECT BLADES

ENTRY SLAVP(IM, ITVU, ITVL)
LOC= 0
NSP= ITVL-ITVU+2
IP = IV(IM)-1
DO 10 IT=1, NSP
IP = IP+1
TSP(IT) = FLOAT(IT+ITVU-1)*HT
10 USP(IT) = U(IP)
USP(NSP) = USP(1)+1.
IP = IV(IM)
INTU = INT(U(IP)*5.)
IF (U(IP).GT.0.) INTU=INTU+1
DO 20 J=1, 5
UINT(J) = FLOAT(INTU)/5.
20 INTU = INTU+1
UINT(6) = UINT(1)
GO TO 100

C SLAVBB CALCULATES ALONG VERTICAL MESH LINES WHICH INTERSECT BLADES
ENTRY SLAVBB(IM)
LOC = 1
ITVUPL = ITV(IM, 1)
ITVLM1 = ITV(IM, 2)
ITVU = ITVUPL - 1
ITVL = ITVLM1 + 1
NSP = ITVLM - ITVU + 1
TSP(1) = ITV(IM, 1)
TSP(NSP) = ITV(IM, 2)
USP(1) = BV(1)
USP(NSP) = BV(2)
IP = IV(IM) - 1
NSPMI = NSP - 1
IFI2.GT.NSPMI GO TO 70
DO 60 IT = 2, NSPMI
IP = IP + 1
TSP(IT) = FLOAT(IT + ITVU - 1) * HT
60 USP(IT) = U(IP)
70 DO 80 I = 1, 6
80 UINT(I) = FLOAT(I - 1) / 5.
C
C FOR BOTH SLAVP AND SLAVBB, CALCULATE RHO*W-SUB-M IN THE REGION, AND
C RHO*W AT VERTICAL MESH LINE INTERSECTIONS ON THE BLADE SURFACES
C
100 CALL SPLINE(TSP, USP, NSP, DUDT, AAA)
   IT = LOC
   IPU = IV(IM)
   IPL = IV(IM) + 1 - 1
   DO 110 IP = IPU, IPL
   IT = IT + 1
110 RWM(IP) = DUDT(IP) * WTFL / BE(1IM) / RM(IM)
120 IF (LOC.EQ.0) GO TO 130
   WMB(IM, 1) = DUDT(1) * WTFL / BE(IM) / RM(IM)
   WMB(IM, 2) = DUDT(NSP) * WTFL / BE(IM) / RM(IM)
   RMDTU2 = (RM(IM) * DTDMV(IM, 1))**2
   RMDTL2 = (RM(IM) * DTDMV(IM, 2))**2
   IF (RMDTU2.GT.10000.) WMB(IM, 1) = 0.
   IF (RMDTL2.GT.10000.) WMB(IM, 2) = 0.
   WMB(IM, 1) = ABS(WMB(IM, 1)) * SQRT(1. + RMDTU2)
   WMB(IM, 2) = ABS(WMB(IM, 2)) * SQRT(1. + RMDTL2)
130 IF (SLCRD.LE.0) RETURN
   NI = 6
   CALL SPLINT(USP, TSP, NSP, UINT, NI, TINT, AAA)
   DO 140 J = 1, 6
   L = (J - 1) * MM + IM
140 TSL(L) = TINT(J)
   IF (IM.EQ.1) WRITE(6, 1000)
   WRITE(6, 1010) MV(IM), (UINT(J), TINT(J), J = 1, 6)
RETURN
1000 FORMAT(1H1/30X, 2HSTREAM LINE COORDINATES/17HLM
   1 3(7X,10HSTREAM FN., 10X,5HTHETA, 4X)//)
1010 FORMAT(1X, 7G18.7/19X, 6G18.7))
END
SUBROUTINE TANG

TANG CALCULATES RHOM*W-SURF-THETA AND THEN RHOM*W THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND CALCULATES THE VELOCITY ANGLE, BETA, THROUGHOUT THE REGION.

COMMON SRW, ITER, IFND, LER(2), NER(2)
COMMON /AKRPHG/ A(2500,4), U(2500), K(2500), RH(2500)
COMMON /INPGAM/, AK, TIP, RHOP, WFL, OMEGA, ORF, BETA1, PETA0,
1 MB1, MGO, WM, NBDI, NL, NRS, PR(50), RSP(50), RESP(50),
2 HDAT, ANDK, ESROR, STRFN, SLCRD, INTVL, SURVL
COMMON /CALCON/ MBO, MBI, MBOMI, MBOPI, MMM, HM, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, TWW, CPTIP, TGROG, TDI, THO, LAMBA, THL, ITMIN, ITMAX,
2 NIP, IMS(21), O(2), MV(100), TV(100), TV(100), TV(100),
3 ETMV(100,2), PETA(100,2), MH(100,2), DTMV(100,2), RTHA(100,2),
4 RM(100,2), REF(100,2), RM(100), BE(100), DBM(100), SAL(100),
5 AAA(100)
DIMENSION SPM(100), USP(100), OUMM(100)
DIMENSION W(2500), WMB(2500), RMB(100,2), WT(100,2),
1 XDOWN(400), YACROS(400)
EQUIVALENCE (AI(1,1), W(1)), (AI(1,2), WMB(1)), (AI(1,3), PETA1),
1 (AI(4), RMB(1)), (AI(201,4), WMB(1)), (AI(401,4), XDOWN(1))
2 (K1, YACROS1)
INTEGER RLDAT, AANDK, ESROR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
1 UPPE, SI, ST, SRW
REAL K, KAK, LAMBA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIPI
EXTERNAL LLI, BLDAT

PERFORM CALCULATIONS ALONG ONE HORIZONTAL LINE AT A TIME

IT = ITMIN
10 IF (IT.GT. ITMAX) RETURN
SI = 0

ON THE GIVEN HORIZONTAL MESH LINE, FIND A FIRST POINT IN THE REGION

IF (IT.GE.0 AND IT.LT. NBDI) GO TO 60
IM = IM1
20 IM = IM1 + 1
IF (IM.GT. MBO) GO TO 200
SURF = 1
IF (IT.GE. TV(IM,1), AND IT.LT. TV(IM-1,1)) GC TO 70
IF (IM.EQ. MBO1, AND IT.EQ. TV(MBO1,1), AND TV(MBO,1) = TV(MBO,2)
1 +NBD1, EQ.2) GC TO 70
SURF = 2
IF (IT.LT. TV(IM,2), AND IT.GT. TV(IM-1,2)) GO TO 70
GO TO 20

FIRST POINT IS ON BOUNDARY A-H

60 IM = 1
IM = 1
SPW(1) = MV(1)
USP (1) = U(IT+1)

41
GO TO 90

C FIRST POINT IS ON A BLADE SURFACE

70 SI = SURF
IM1 = IM-1
IM2 = IM
TH = FLOAT(IT)*HT
MVIM1 = MV(IM1)
IF (IM.EQ.IMT) MVIM1 = MV(IM1)*(MV(IM2)-MVIM1)/1000.
LEK(2) = 5
C BLCC (VIA ROOT) CALL NO. 5
IF (SI.LE.1.AND.IM1.NE.MBI) CALL ROUT(MVIM1,MV(IM2),TH,BL1,DTLR,
1 ANS,AAA)
LEK(2) = 6
C BLCC (VIA ROOT) CALL NO. 6
IF (SI.LT.Z) CALL ROUT(MVIM1,MV(IM2),TH,BL2,DTLR,ANS,AAA)
IF (SI.LE.1.AND.IM1.EQ.MBI) ANS = MV(MBI)
SPW(IM1) = ANS
USP(IM1) = BV(SI)

C MOVE ALONG HORIZONTAL MESH LINE UNTIL MESH LINE INTERSECTS BOUNDARY

90 IF(IM.LT.MBI.OR.IM.GT.MBI) GO TO 120
SURF = 1
IF (IT.LT.ITV(IM,SURF).AND.IT.GT.ITV(IM-1,SURF)) GO TO 140
SURF = 2
IF (IT.LE.ITV(IM,SURF).AND.IT.LE.ITV(IM-1,SURF)) GO TO 140
120 SPW(IM) = MV(IM)
IP = IPF(IM,IT)
USP(IM) = U(IP)
IF (IM.EQ.MM) GO TO 130
IM = IM+1
GO TO 90

C FINAL POINT IS ON BOUNDARY D-F

130 IMT = MM
GO TO 150

C FINAL POINT IS ON A BLADE SURFACE

140 ST = SURF
IMT=IM
IMT1= IMT-1
TH = FLOAT(IT)*HT
MVIM1 = MV(IMT1)
IF (IMT1.EQ.MM) MVIM1 = MVIM1*(MV(IM2)-MVIM1)/1000.
LEK(2) = 7
C BLCC (VIA ROOT) CALL NO. 7
IF (ST.LE.1.AND.IMT.NE.MBI) CALL ROUT(MVIM1,MV(IMT),TH,BL1,
1 DTLR,ANS,AAA)
LEK(2) = 8
C BLCC (VIA ROOT) CALL NO. 8
IF (ST.LE.2) CALL ROUT(MVIM1,MV(IMT),TH,BL2,DTLR,ANS,AAA)
IF (ST.LE.1.AND.IMT.EQ.MBI) ANS = MV(MBI)
CALCULATE RHO*W-SUB-THETA AND THEN RHO*W AND BETA IN THE REGION

150 NSP= |IP-IM1|+1
CALL SPLINE(SPNI(IP1),USPI(IP1),NSP,DUO(IP1),AAA(IP1))
FIRST=1
IF (IP1.NE.1) FIRST=IM2
LAST= IM1
IF (IP1.NE.IM1) PLEASE WAIT
IF (FIRST.GT.LAST) PLEASE WAIT
DO 10 I=FIRST,LAST
RWT = -DUO(IP1)*WTFL/BEH(IP1)
IP = INT(IP1)
W(IP) = SQRT(RNH**2+RWM(IP)**2)
160 ETA(IP) = ATAN2(RNH,RWM(IP))*57.295779
CALCULATE RHL;_W
OF _IHF BLADE SURFACES

170 IF (IM1.F0.1) PLEASE WAIT
CALL SEARCH (SPNI(IP1),ST,IHS)
ANS = -DUO(IP1)*WTFL/BEH(IHS,ST)
W(IP,ST) = SQRT(RNH**2+RWM(IP,ST)**2)
180 IF (IM1.F0.1) PLEASE WAIT
CALL SEARCH (SPNI(IP1),ST,IHS)
ANS = -DUO(IP1)*WTFL/BEH(IHS,ST)
W(IP,ST) = SQRT(RNH**2+RWM(IP,ST)**2)
190 GO TO 200
200 IT = IT+1
GO TO 10

$IBFTC SEARCH DEBUG
SUBROUTINE SEARCH (DIST,SURF,IS)
SEARCH LOCATES THE POSITION OF A GIVEN VALUE OF M IN THE MH ARRAY
COMMON /ICALCON/MBIM1,MBIP1,MBDM1,MBDP1,MMM1,HM1,HT,DTLR,DMLR,
PITCH,CP,EXPON,TW,LAMBDATW,ITMIN,ITMAX,
NIP,IMSI21,B¥(21,M¥CIOOI,IVIIOll,ITVI100,21,TVllOO,21,
DTO¥IIOO,21,BETAVIIOO,21,MHIIOO,21,OTDMHCIOO.21,BETAH(100,21,
RHH(100,2),BEHI100,2),RMH(100,2),BE(100),DBDMH(100),SAL(100),
AAA(100)
INTEGER BLOAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
UPPER,ST,SRW
REAL K,KAK,LAMBDATLMAX,ME,MLE,MR,MSL,MP,MM,MBIM1
DO 10 I=1,ITMAX
IF (ABS(MH1(SURF)-DIST).GT.DMLR) GO TO 10
IS = I
RETURN
10 CONTINUE
WRITE (6,1000) DIST,SURF
STOP
1000 FORMAT (38HL SEARCH CANNOT FIND M IN THE MH ARRAY/TH DIST =,G14.6,
110X,6HSURF =,G14.6)
END
SUBROUTINE VELOCY

VELOCY CALLS SUBROUTINES TO CALCULATE DENSITIES AND VELOCITIES THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND IT PLOTS THE SURFACE VELOCITIES.

COMMON /AUKRHO/A(2500,4),U(2500),K(2500),RHO(2500)
COMMON /INP/GAM,AR,TIP,ROHIOPT,WTL,OMEGA,ORF,BETA1,BETA0,
  1 MBI,MBO,MM,NBBI,NBL,NRP,MR(50),RMS(50),BES(50),
  2 BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
COMMON /CALCON/MBI1,MBO1,MBM1,MBP1,MR1,MSP1,MV1,MV1,MV1
  1 PITCH,CP,EXPON,TWW,CPTIP,TRGROG,TBI,TBO,LABDA,TLW,ITMIN,ITMAX,
  2 IMS1,IV(100),IV(100),TV(100),TV(100),

DATA KKK(4),/1H/=K,AAA(100)
DIMENSION KKK(10),W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
  1 XDOWN(400),YACROS(400)
EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
  2 (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
  3 (K(1),YACROS(1))
INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
  1 UPPER,S1,ST,SRM
REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
DATA KKK(4)/1H*/,K,-K,=K,AAA(100)
DIMENSION KKK(10)
DO 50 SURF=1,2
NP1 = NP2
IMSS = IMS(SURF)
IF(IMSS.LT.1) GO TO 40
   DO 30 IHS=1,IMSS
      IF(ABS(DTMH(IHS,SURF)*RMH(IHS,SURF)).LT.57735) GO TO 30
      NP1 = WPI+1
      YACROS(NP1) = WTB(IHS,SURF)
      XDOWN(NP1) = MH(IHS,SURF)
   30 CONTINUE
50 KKK(2*SURF+1) = NP1-NP2
C PREPARE INPUT ARRAYS FOR PLOT OF VELOCITIES
C IF(SURVL.LE.0) RETURN
NP2 = 0
C TANGENTIAL COMPONENTS
DO 50 SURF=1,2
NP1 = NP2
IMSS = IMS(SURF)
IF(IMSS.LT.1) GO TO 40
   DO 30 IHS=1,IMSS
      IF(ABS(DTMH(IHS,SURF)*RMH(IHS,SURF)).LT.57735) GO TO 30
      NP1 = WPI+1
      YACROS(NP1) = WTB(IHS,SURF)
      XDOWN(NP1) = MH(IHS,SURF)
   30 CONTINUE
50 KKK(2*SURF+1) = NP1-NP2
50 NP2 = NP1
C MERIDIONAL COMPONENTS
   DO 80 SURF=1,2
      NP1 = NP2
   DO 60 IM=MBIPI,MBOM1
   IF (ABS(DTDMV(IM,SURF)*RM(IM)).GT.1.7321) GO TO 60
      NP1 = NP1+1
      YACROS(NP1) = WMB(IM,SURF)
      XDOWN(NP1) = MV(IM)
   CONTINUE
70 KKK(2*SURF+5) = NP1-NP2
80 NP2 = NP1
C PLOT VELOCITIES
C
   KKK(1) = 1
   KKK(2) = 4
   P = 5.
   WRITE(6,1000)
   CALL PLOTMY(XDOWN,YACROS,KKK,P)
   WRITE(6,1010)
   RETURN
1000 FORMAT(2HPT,50X,24HBLADE SURFACE VELOCITIES)
1010 FORMAT (2HPL,37X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
                 1(M) DOWN THE PAGE /2HPL/
                    2 2HPL,50X,50H+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT/
                    3 2HPL,50X,50H+ - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT/
                    4 2HPL,50X,50HX - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT/
                    5 2HPL,50X,50HO - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT)
END
SUBROUTINE VEL

VEL CALCULATES DENSITIES AND VELOCITIES FROM THE PRODUCT OF DENSITY TIMES VELOCITY

COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON /AUKRHO/ A(2500, 4), U(2500), K(2500), RH0(2500)
COMMON /INP/GAM, AR, TIP, RHOIP, WFL, OMEGA, ORF, BETA, ETAOD,
1 MBI, MB0, MM, ABBI, NBL, NRSP, MR, RMSP, BESP
2 BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, SURVL
COMMON /ICTAN/ MBIM, MBIP1, MBM1, MBM1, HM1, HT, DFLR, DMLR,
1 PITCH, CP, EXPON, FW, CPTIP, TGROG, TBI, TBO, LAMBDA, TWL, ITMIN, ITMAX,
2 NIP, IMS(2), BV(2), MV(100), IV(101), TV(100, 2), TV(100, 2),
3 DTDMH(100, 2), BETAV(100, 2), MV(100, 2), DTDMH(100, 2), BETAV(100, 2),
4 RMB(100, 2), BEM(100, 2), RM(100), BE(100), DB(100), SAL(100),
5 AAA(100)
COMMON /RHOST/RH0HB(100, 2), RH0HB(100, 2)
DIMENSION WCRM(100, 2), WCRM(100, 2), SURFL(100, 2)
DIMENSION W(2500), WRM(2500), BETA(2500), RMSP(100, 2), WTB(100, 2),
1 XDOWN(100), YACROS(400)
EQUIVALENCE (A(1, 1), W(1)), (A(1, 2), WRM(1)), (A(1, 3), BETA(1)),
1 (A(1, 4), WMB(1)), (A(201, 4), WTB(1)), (A(401, 4), XDOWN(1)),
2 (K(1), YACROS(1))

ENTRY VELP(FIRST, LAST)

INTEGER BLDAT, AANDK, ERSOR, STRFN, SLCRD, SURVL, AATEMP, SURF, FIRST,
1 UPPER, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MR, MSL, MV, MVI
IF (FIRST.NE.LAST) RETURN
IF (FIRST.EQ.1.AND.INTVL.GT.0) WRITE(6, 1000)
IF (FIRST.EQ.1) RELER = 0
DO 20 IM = FIRST, LAST
IPU = IV(IM)
IPL = IPU+NBSI-1
TWLMR = 2.*(OMEGA=LAMBDA-(OMEGA*RM(IM))**2
LER(1) = 4
DO 10 IP = IPU, IPL
DENSTY CALL NO. 4
CALL DENSTY(W(IP), RHO(IP), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
10 W(IP) = ANS
IF (INTVL.LE.0) GO TO 20
WRITE (6, 1010) IM, W(IP), BETA(IP), IP = IPU, IPL
20 CONTINUE
RETURN

C VELBB CALCULATES ALONG VERTICAL MESH LINES WHICH INTERSECT BLADES

C
ENTRY VELBB(FIRST, LAST)
IF (FIRST.GT.LAST) RETURN
IF (FIRST.NE.MBI) GO TO 30
SURFL(MBI, 1) = 0.
SURFL(MBI, 2) = 0.
30 DO 70 IM = FIRST, LAST
ITVU = ITV(IM, 1)
ITVL = ITV(IM, 2)
IPUP1 = IPF(IM, ITVU)
IPML1 = IPF(IM, ITVL)
TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RMIIMII)**2
WCR = SQRT(TGRG*TIP*(1.-TWLMR/CPTIP))
IF (ITVL.LT.ITVU) GO TO 50
ALONG THE LINE BETWEEN BLADES
LER(1) = 5
DO 40 IP = IPUP1, IPML1
C DENSITY CALL NO. 5
CALL DENSITY(W(IP), RHO(IP), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
40 W(IP) = ANS
IF (INTVL.LE.0.0) GO TO 50
WRITE (6, 1010) IM, (W(IP), BETA(IP), IP = IPUP1, IPML1)
C ON THE UPPER SURFACE
50 RHOB = RHOVB(IM, 1)
LER(1) = 6
C DENSITY CALL NO. 6
CALL DENSITY(WMB(IM, 1), RHOVB(IM, 1), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
WMB(IM, 1) = ANS
WWCRM(IM, 1) = WMB(IM, 1)/WCR
IF (IM.EQ.MBI) GO TO 60
DELTV = TV(IM-1, 1) - TV(IM, 1)
SURFL(IM, 1) = SURFL(IM-1, 1) + SQRT((MV(IM) - MV(IM-1))**2 + 1 * (DELTV**2 + RM(IM) + RM(IM-1))/2.)**2)
60 RELER = AMAX1(RELER, ABS(RHOB - RHOVB(IM, 1))/RHOVB(IM, 1)))
C ON THE LOWER SURFACE
RHOB = RHOVB(IM, 2)
LER(1) = 7
C DENSITY CALL NO. 7
CALL DENSITY(WMB(IM, 2), RHOVB(IM, 2), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
WMB(IM, 2) = ANS
WWCRM(IM, 2) = WMB(IM, 2)/WCR
IF (IM.EQ.MBI) GO TO 70
DELTV = TV(IM-1, 2) - TV(IM, 2)
SURFL(IM, 2) = SURFL(IM-1, 2) + SQRT((MV(IM) - MV(IM-1))**2 + 1 * (DELTV**2 + RM(IM) + RM(IM-1))/2.)**2)
70 RELER = AMAX1(RELER, ABS(RHOB - RHOVB(IM, 2))/RHOVB(IM, 2)))
RETURN
C
C VELSUR CALCULATES ALONG A BLADE SURFACE
C
ENTRY VELSUR
DO 90 SURF = 1, 2
IMSS = IMS(SURF)
IF (IMSS.EQ.0) GO TO 90
DO 80 IHS = 1, IMSS
TWLMR = 2.*OMEGA*LAMBD-(OMEGA*RHOH(IHS,SURF)**2)
WCR = SQRT(TGROG_TIP*(1.-TWLMR/CPTIP))
RHOB = RHOHB(IHS,SURF)

C DENSTY CALL NO. 8
C CALL DENSTY(WTB(IHS,SURF),RHOHB(IHS,SURF),ANS,TWLMR,CPTIP,
1. EXPON,RHOIP,GAM_AR,TIP)
WTS(IHS,SURF) = ANS
WWCR(IHS,SURF) = WTB(IHS,SURF)/WCR
80 RELER = AMAX1(RELER,ABS((RHOB-RHOHB(IHS,SURF))/RHOHB(IHS,SURF)))
90 CONTINUE
IF (RELER.LT.0.01) IEND = IEND+1
WRITE(6,1080) ITER,RELER

C WRITE ALL BLADE SURFACE VELOCITIES
C IF (SURVL.LE.0) RETURN
WRITE(6,1020)
WRITE(6,1040) (MV(IM),WMB(IM,1),BETAV(IM,1),SURFL(IM,1),
1. WWCRM(IM,1),WMB(IM,2),BETAV(IM,2),SURFL(IM,2),WWCRM(IM,2),
2. IM=MBI,MBO)
WRITE(6,1050)
DO 100 SURF=1,2
IMSS = IMS(SURF)
IF (IMSS.LT.1) GO TO 100
WRITE(6,1060) SURF
WRITE(6,1070) (MH(IHS,SURF),WTB(IHS,SURF),BETAH(IHS,SURF),
1. WWCR(IHS,SURF), IHS=1,IMSS)
100 CONTINUE
RETURN
1000 FORMAT(1HI/40X,34HVELOCITIES AT INTERIOR MESH POINTS )
1010 FORMAT(1HI/16X,3HI=-,3HI=5(24H VELOCITY ANGLE(DEG)) /
1. 15X,5(G15.4,F9.2))
1020 FORMAT(1HI/16X,1H*,18X,49HSURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS)
1. 18X,1H*/16X,1H*,53X,1H*/16X,1H*,19X,15HBLADE SURF 2FACE 1,19X,1H*,20X,15HBLADE SURF 2,18X,1H*/7X,1HM,8X,1H*,2(3X,
3. 3BHVELOEACY,3X,23HANGLE(DEG) SURF. LENGTH,5X,5HW/WCR,6X,1H*)
1040 FORMAT(1HI,G13.4,3H *,2(G12.4,F9.2,2G15.4,3H *))
1050 FORMAT(1HI/3X,49HSURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS)
1060 FORMAT(//22X,15HBLADE SURFACE,11/7X,1HM,10X,8HVELOEACY,3X,1OHANG
1. 1LE(DEG),3X,5HW/WCR)
1070 FORMAT(1HI,2G13.4,F9.2,2G15.4)
1080 FORMAT(14HITERATION NO.,3X,36HMAXIMUM RELATIVE CHANGE IN DENSITY =,G11.4)
END
$IBFTC$ SPLINE DEBUG

SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)

C
C SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
C ADJACENT POINT
C
COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
1 G(200)
DIMENSION X(N),Y(N),EM(N),SLOPE(N)
INTEGER Q
DO 10 I=2,N
10 S(I)=X(I)-X(I-1)
NO=N-I
IF(NO.LT.2) GOTO 30
DO 20 I=2,NO
A(I)=S(I)/6.
B(I)=(S(I)+S(I+1))/3.
C(I)=S(I+1)/6.
20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
30 A(N) = -.5
B(I)=1.
B(N)=1.
C(I) = -.5
F(I)=0.
F(N)=0.
W(I)=B(I)
SB(I)=C(I)/W(I)
G(I)=0.
DO 40 I=2,N
W(I)=B(I)-A(I)*SB(I-1)
SB(I) = C(I)/W(I)
40 G(I) = (F(I)-A(I)*G(I-1))/W(I)
EM(N)=G(N)
DO 50 I=2,N
50 EM(K)=G(K)-SB(K)*EM(K+1)
SLOPE(I)=-S(2)/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/S(2)
DO 60 I=2,N
60 SLOPE(I)=-S(2)/6.*(2.*EM(I-1)+EM(I-2))+(Y(I)-Y(I-1))/S(I)
IF (Q.EQ.13) WRITE(6,1000) N,X(I),Y(I),SLOPE(I),EM(I),I=1,N)
RETURN
1000 FORMAT (2X,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
12HEM/(4F20.8))
END
SUBROUTINE SPLINT (X,Y,N,L,M,YINT,DYDX)

SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
FOR A SPLINE CURVE
END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
ADJACENT POINT

COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
G(100),EM(100)
DIMENSION X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX)
INTEGER Q
IF(MAX.LE.0) RETURN
Q = 0
DO 10 I=2,N
10 S(I)=X(I)-X(I-1)
NO=N-1
IF(NO.LE.2) GO TO 30
DO 20 I=2,NO
A(I)=S(I)/6.0
B(I)=(S(I)+S(I+1))/3.0
C(I)=S(I+1)/6.0
20 F(I)=(Y(I+1)-Y(I))/(S(I+1)-(Y(I)-Y(I-1))/S(I))
30 A(N) = -.5
B(N)=1.0
C(N) = -.5
F(N)=0.0
F(N)=0.0
W(I)=B(I)
SB(I)=C(I)/W(I)
G(I)=0.0
DO 40 I=2,N
W(I)=B(I)-A(I)*SB(I-1)
SB(I)=C(I)/W(I)
40 G(I)=(F(I)-A(I)*G(I-1))/W(I)
EM(N)=G(N)
DO 50 I=2,N
50 K=N+I-1
50 EM(K)=G(K)-SB(K)*EM(K+1)
DO 140 I=1,MAX
K=2
140 IF(Z(I)-X(I)) 70,60,90
70 IF(Z(I).GE.(1.1*X(I)-.1*X(2))) GO TO 120
WRITE (6,10000) Z(I)
Q = 16
GO TO 120
80 K=N
IF(Z(I).LE.(1.1*X(N)-.1*X(N-1))) GO TO 120

50
WRITE (6,1000) Z(I)
Q = 16
GO TO 120
90 IF(Z(I)-X(K)) 120,100,110
100 YINT(I)=Y(K)
GO TO 130
110 K=K+1
IF(K=N) 90,90,80
120 YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6.*S(K)+EM(K)*Z(I)-X(K-1))**3/6.*
1/S(K)+(Y(K)/S(K)-EM(K)*S(K)/6.)*(Z(I)-X(K-1))+(Y(K-1)/S(K)-EM(K-1)
2*S(K)/6.)*(X(K)-Z(I))
130 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/S(K)+EM(K)*(X(K-1)-Z(I))**2/2.
10/S(K)+(Y(K)-Y(K-1))/S(K)-(EM(K)-EM(K-1))*S(K)/6.0
140 CONTINUE
MXA = MAX(MAX,MAX)
IF(Q.EQ.16) WRITE(6,1010) N,MAX,(X(I),Y(I),Z(I),YINT(I),DYDX(I),,
1=1,MAX)
Q = III
RETURN
1000 FORMAT (54H SPLINT USED FOR EXTRAPOLATION. EXTRAPOLATED VALUE = ,
16H14.6)
1010 FORMAT (2X,21HNQ. OF POINTS GIVEN =,,I3,30H, NO. OF INTERPOLATED PO
1 LINTS =,,I3,10X,1HX,19X,1HY,16X,11HX-INTERPOL.,9X,11HY-INTERPOL.,
28X,14HDYDX-INTERPOL./(5E20.8))
END
SUBROUTINE SPLN2Z (X,Y,YIP,YNP,N,SLOPE,EM)

SPLN2Z CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS

COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
1 G(200)
DIMENSION X(N),Y(N),EM(N),SLOPE(N)
INTEGER Q

DO 10 I=2,N
10 S(I)=X(I)-X(I-1)

NO=N-1
IF (NO.LT.2) GO TO 30

DO 20 I=2,NO
A(I)=S(I)/6.
B(I)=(S(I)+S(I+1))/3.
C(I)=S(I+1)/6.
20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)

30 A(1)=S(N)/6.
B(1)=S(N)/3.
C(1)=S(N)/6.
F(N)=YNP-(Y(N)-Y(N-1))/S(N)

W(1)=B(1)
SB(1)=C(1)/W(1)
G(1)=F(1)/W(1)

DO 40 I=2,N
W(I)=B(I)-A(I)*SB(I-1)
SB(I)=C(I)/W(I)
40 G(I)=(F(I)-A(I)*G(I-1))/W(I)

EM(1)=5INI
DO 50 I=2,N
EM(I)=G(I)-SB(K)*EM(K+1)
50 SLOPE(I)=-S(2)/6.*E(2)*EM(I)+EM(I)+E(2))+(Y(2)-Y(1))/S(2)

DO 60 I=2,N
60 SLOPE(I)=S(I)/6.*E(2)*EM(I)+EM(I)+E(2))+(Y(I)-Y(I-1))/S(I)

IF (Q.EQ.18) WRITE(6,1000) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
RETURN

1000 FORMAT (2X,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
12HEM/(4F20.8))
END
SUBROUTINE ROOT(A,B,Y,FUNCTION,FUNCT,TOLERY,X,DFX)

ROOT FINDS A ROOT FOR (FUNCTION - Y) IN THE INTERVAL (A,B)

COMMON SRW,ITER,IER(2),NER(2)
INTEGER SRW

IF (SRW.EQ.2) WRITE(6,100) A,B,Y,TOLERY
TOLERX= (B-A)/1000.
AB2= (A+B)/2.
I= 0
X= A

10 CALL FUNCTION(X,FX,DFX,INF)
IF (SRW.EQ.2) WRITE(6,1010) I,X,FX,DFX,INF
IF (ABS(Y-FX).LT.TOLERY) RETURN
IF (I.GE.1000) GO TO 30
I= I+1
IF (INF.NE.O .OR. DFX.EQ.O.) GO TO 20
X= (Y-FX)/DFX
IF (X.GE.A .AND. X.LE.B) GO TO 10
X= A+TOLERX*FLOAT(I)
IF(I.EQ.I) X= B
GO TO 10

20 IF (X.LT.AB2) X= X+TOLERX
IF (X.GE.AB2) X= X-TOLERX
GO TO 10

30 WRITE(6,1020) LER(2),A,B,Y
STOP

1000 FORMAT (13HINPUT ARGUMENTS FOR ROOT -- A =G13.5,3X,3H3 =,G13.5,
1 3X,3HY =,G13.5,3X,8HTOLERY =,G13.5/17H ITER. NO.  X,17X,
2 2HFX,15X,3HDFX,10X,3HINF)
1010 FORMAT (5X,13,G16.5,2G18.5,16)
1020 FORMAT (14HLROOT CALL NO.,13/47H ROOT HAS FAILED TO CONVERGE IN 10
100 ITERATIONS/4H A =,G14.6,10X,3H3 =,G14.6,10X,3HY =,G14.6)
END
SUBROUTINE BLCD

C BLCD CALCULATES BLADE THETA COORDINATE AS A FUNCTION OF M

COMMON SRW, ITER, IEND, LER(2), NER(2)
COMMON INP/GAM, AR, TIP, RHOIP, WFL, OMEGA, ORF, BETAI, BEIAO,
1 MBI, MBO, MM, NBI, NBL, NRS, MR(50), RSP(50), BESP(50),
2 BLDAT, ANDK, ESROR, STRFN, SLCD, INTVL, SURVL
COMMON /CALCONE/MBI1, MB11, MBM1, MM11, H1, HT, DTLR, DMLR,
1 PITCH, CP, EXPON, TW, CPTIP, TGRO, TI, TBO, LAMDA, TWL, ITMIN, ITMA,
2 NIP, IMS(2), TV(2), MV(100), TV(101), TV(100, 2), TV(100, 2),
3 DTDM(100, 2), BETAV(100, 2), MHI(100, 2), BETAH(100, 2),
4 RMH(100, 2), BEH(100, 2), RM(100), BET(100), BHM(100), SAL(100),
5 AAA(100)
COMMON /GEOMIN/ CHORD(2), STGR(2), MLE(2), THLE(2), RMI(2), RMS(2),
1 R1(2), R0(2), BET(2), BETO(2), NSP(2), MSP(50, 2), THSP(50, 2)
COMMON /BLCDON/ EM(50, 2), INIT(2)
ENTRY BL1(M, THETA, DTDM, INF)

INTEGER BLDAT, ANDK, ESROR, STRFN, SLCD, SURVL, AATEMP, SURF, FIRST,
1 UPPER, SI, ST, SRW
REAL K, KAK, LAMDA, LMAX, MH, MLE, MR, MSL, MSP, MV, MVIM
REAL M, MMLE, MSPM, MMMSP
SURF= 1
SIGN= 1.
GO TO 10
ENTRY BL2(M, THETA, DTDM, INF)
SURF= 2
SIGN= -1.
10 INF= 0
NSP= NSPI(SURF)
IF (INIT(SURF).EQ.13) GO TO 30
INIT(SURF)= 13
C INITIAL CALCULATION OF FIRST AND LAST SPLINE POINTS ON BLADE

AA = BETI(SURF)/57.295779
AA = SIN(AA)
MSP1(SURF) = RI(SURF)*(1.-SIGN*AA)
BB = SQRT(1.-AA**2)
THSP1(SURF) = SIGN*BB*RI(SURF)/RMI(SURF)
BET1(SURF) = AA/BB/RMI(SURF)
AA = BET0(SURF)/57.295779
AA = SIN(AA)
MSPINS(SURF) = CHORD(SURF)-ROI(SURF)*(1.+SIGN*AA)
BB = SQRT(1.-AA**2)
THSPINS(SURF) = STGR(SURF)+SIGN*BB*ROI(SURF)/RMO(SURF)
BETO(SURF) = AA/BB/RMO(SURF)
DO 20 IA=1, NSP
MSP(IA, SURF)= MSP1(A, SURF)+MSPINS(A, SURF)
20 THSP1(A, SURF)= THSP1(A, SURF)+THLE(SURF)
CALL SPL22(MSP(1, SURF), THSP(1, SURF), BETI(SURF), BETO(SURF), NSP,
1 AAA, EM1(SURF))
IF (BLDAT.LE.0) GO TO 30
IF (SURF.EQ.1) WRITE(16, 1000)
WRITE(16, 1010) SURF
WRITE (6, 1020) (MSP1(IA, SURF), THSP(IA, SURF), AAA(IA), EM1(IA, SURF),
1 IA=1, NSP)
C BLADE COORDINATE CALCULATION
30 KK = 2
IF (M.GT.\$P(i,SURF)) GO TO 50
C
C AT LEADING EDGE RADIUS
C
MMLE = M-MLE(SURF)
IF (MMLE.LT.-DMLR) GO TO 90
MMLE = AMAX1(0.,MMLE)
THETA = SQRT(MMLE*(2.*RI(SURF)-MMLE))*SIGN
IF (THETA.EQ.0.) GO TO 40
RMM = RI(SURF)-MMLE
DTDM = RMM/THETA/RMI(SURF)
THETA = THETA/RMI(SURF)+THLE(SURF)
RETURN
40 INF = 1
DTDM = 1.E10*SIGN
THETA = THLE(SURF)
RETURN
C
C ALONG SPLINE CURVE
C
50 IF (M.LE.$P(KK,SURF)) GO TO 60
IF (KK.GE.NSP) GO TO 70
KK = KK+1
GO TO 50
60 S = $P(KK,SURF)-$P(KK-1,SURF)
EMKM = EMIK-1,SURF)
EMK = EM(KK,SURF)
MMSPM = M-\$P(KK-1,SURF)
THK = THS(KK,SURF)/S
THKM = THS(KK-1,SURF)/S
THETA = EMK*MSPMM**3/6./S + EMK*MMMSP**3/6./S + (THK-EMK*S/6.)*
+ (THK-THKM-EMK*SIGN)
DTDM = -EMK*MSPMM**2/2./S + EMK*MMMSP**2/2./S + THK-THKM-EMK*SIGN
RETURN
70 C AT TRAILING EDGE RADIUS
C
C C CM= CHORD(SURF)+MLE(SURF)-M
IF (CM.MLE.-DMLR) GO TO 90
CM= AMAX1(0.,CM){
THETA = SQRT(CMM*(2.*SD(SURF)-CM)*SIGN
IF (THETA.EQ.0.) GO TO 80
RMM = SD(SURF)-CM
DTDM = -RMM/THETA/RMD(SURF)
THETA = STGR(SURF)+THETA/RMD(SURF)+THLE(SURF)
RETURN
80 INF = 1
DTDM = 1.E10*SIGN
THETA = THLE(SURF)+STGR(SURF)
RETURN
C
C ERROR RETURN
C
90 WRITE(6,1030) \$R(2),M,SURF
STOP
1000 FORMAT (1HL,13X,33HBLADE DATA AT INPUT SPLINE POINTS)
1010 FORMAT(1HL,17X,16HBLADE SURFACE,14)
1020 FORMAT (7X,I1H,10X,5HTHETA,10X,10HDERIVATIVE,5X,10H2ND DERIV. / 1 L (4G15.5) )
1030 FORMAT (14HLBLCDO CALL NO.*13/33H M COORDINATE IS NOT WITHIN BLADE/ 14H M =,G14.6,10X,6HSURF =,G14.6)
END
SUBROUTINE DENSTY(RHOW, RHO, VEL, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)

DENSTY CALCULATES DENSITY AND VELOCITY FROM THE WEIGHT FLOW PARAMETER DENSITY TIMES VELOCITY

COMMON SRW, ITER, IEND, LER(2), NER(2)

VEL = RHOW / RHO
IF (VEL.NE.0.) GO TO 10
RHO = RHOIP
RETURN

10 TTIP = 1.-(VEL**2+TWLMR)/CPTIP
IF(TTIP.LT.0.) GO TO 30
TEMP = TTIP**(*EXPON-1.)
RHT = RHOIP*TEMP*TTIP
RHOWP = -VEL**2/GAM*RHOIP/AR*TEMP/TIP*RHOT
IF(RHOWP.LE.0.) GO TO 30
VELNEW = VEL-(RHOW-RHOT*VEL)/RHOWP
IF(ABS(VELNEW-VEL)/VEL.LE.00001) GO TO 20
VEL = VELNEW
GO TO 10

20 VEL = VELNEW
RHO = RHOW/VEL
RETURN

30 TGROG = 2.*GAM*AR/(GAM+1.)
VEL = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
RWMORW = RHOW/RHO/VEL
NER(1) = NER(1)+1
WRITE(6,1000) LER(1), NER(1), RWMORW
IF(NER(1).EQ.50) STOP
RETURN

1000 FORMAT(16HLDENSITY CALL NO., 13/9H NER(1) =, 13/10H RHO*W IS , F7.4, 134H TIMES THE MAXIMUM VALUE FOR RHO*W)
END

FUNCTION IPF(IM, IT)

COMMON ICALCON/MBIN1, MBIP1, MBOM1, MBDP1, MMML, HML, HT, DTLR, DMLR, 1 PITCH, CP, EXPON, TWM, CPTIP, TGRDG, TBI, TB0, LAMBDA, TWL, ITMIN, ITMAX, 2 MIP, IMS(2), BV(2), MV(100), IV(101), ITV(100,2), TV(100,2), 3 DTOMV(100,2), BEHAV(100,2), MH(100,2), DTOMH(100,2), BEH(100,2), 4 RPH(100,2), BHEH(100,2), RM(100), BE(100), D8DM(100), SAL(100), 5 AAAA,100
IPF = IV(IM)+IT-ITV(IM,1)
RETURN
END

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, December 9, 1968,
491-05-00-02-22.
REFERENCES


