FORTRAN PROGRAM FOR CALCULATING VELOCITIES AND STREAMLINES ON THE HUB-SHROUD MID-CHANNEL FLOW SURFACE OF AN AXIAL- OR MIXED-FLOW TURBOMACHINE

II - Programmer's Manual

by Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1974
**Abstract**

A FORTRAN-IV computer program, MERIDL, has been developed that obtains a subsonic or shock-free transonic flow solution on the hub-shroud mid-channel flow surface of a turbomachine. The blade row may be fixed or rotating and may be twisted and leaned. Flow may be axial or mixed, up to $45^\circ$ from axial. Upstream and downstream flow variables can vary from hub to shroud, and provision is made to correct for loss of stagnation pressure. The results include velocities, streamlines, and flow angles on the flow surface and approximate blade surface velocities. Subsonic solutions are obtained by a finite-difference stream-function solution. Transonic solutions are obtained by a velocity-gradient method, using information from a finite-difference stream-function solution at a reduced mass flow.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>OVERALL PROGRAM PROCEDURE</td>
<td>4</td>
</tr>
<tr>
<td>DETAILED PROGRAM PROCEDURE</td>
<td>8</td>
</tr>
<tr>
<td>STORAGE REQUIREMENTS</td>
<td>12</td>
</tr>
<tr>
<td>CONVENTIONS USED IN PROGRAM</td>
<td>12</td>
</tr>
<tr>
<td>LABELED COMMON BLOCKS</td>
<td>14</td>
</tr>
<tr>
<td>MAIN PROGRAM</td>
<td>15</td>
</tr>
<tr>
<td>SUBROUTINES</td>
<td>15</td>
</tr>
<tr>
<td>Subroutine INPUT</td>
<td>15</td>
</tr>
<tr>
<td>Subroutine INPLOT</td>
<td>15</td>
</tr>
<tr>
<td>Subroutine MESHO</td>
<td>16</td>
</tr>
<tr>
<td>Subroutine CROSICD</td>
<td>19</td>
</tr>
<tr>
<td>Subroutine PRECAL</td>
<td>20</td>
</tr>
<tr>
<td>Subroutine THETOM</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine THIKOM</td>
<td>22</td>
</tr>
<tr>
<td>Subroutine LOSSOM</td>
<td>23</td>
</tr>
<tr>
<td>Subroutine MEPLIT</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine PTBDRY</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine VBDRY</td>
<td>25</td>
</tr>
<tr>
<td>Subroutine INIT</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine COEF</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine SOR</td>
<td>28</td>
</tr>
<tr>
<td>Subroutine NEWRHO</td>
<td>29</td>
</tr>
<tr>
<td>Subroutine OUTPUT</td>
<td>33</td>
</tr>
<tr>
<td>Subroutine BLDVEL</td>
<td>38</td>
</tr>
<tr>
<td>Subroutine ILETE</td>
<td>39</td>
</tr>
<tr>
<td>Subroutine TSONIN</td>
<td>40</td>
</tr>
<tr>
<td>Subroutine INDEV</td>
<td>40</td>
</tr>
<tr>
<td>Subroutine SLPLOT</td>
<td>41</td>
</tr>
<tr>
<td>Subroutine SVPLOT</td>
<td>41</td>
</tr>
<tr>
<td>Subroutine TVELCY</td>
<td>41</td>
</tr>
</tbody>
</table>

**PRECEDING PAGE BLANK NOT FILMED**
A FORTRAN-IV computer program, MERIDL, has been developed which obtains a subsonic or transonic, nonviscous flow solution on the hub-shroud mid-channel flow surface of a turbomachine. The flow must be essentially subsonic, but there may be locally supersonic flow. The solution is for two-dimensional, adiabatic shock-free flow. The blade row may be fixed or rotating and may be twisted and leaned. The flow may be axial or mixed, up to approximately 45° from axial. Upstream and downstream flow conditions can vary from hub to shroud, and provision is made for an approximate correction for loss of stagnation pressure.

The basic analysis is based on the stream function and consists of the solution of the simultaneous, nonlinear, finite-difference equations of the stream function. This basic solution, however, is limited to strictly subsonic flow. When there is locally supersonic flow, a transonic solution must be obtained. The transonic solution is obtained by a combination of a finite-difference stream-function solution and a velocity-gradient solution. The finite-difference solution at a reduced mass flow provides information which is used to obtain a velocity-gradient solution at the full mass flow.

The program input consists of blade and flow channel geometry, upstream and downstream flow conditions from hub to shroud, and mass flow. The output includes streamline coordinates, flow angles, and velocities on the mid-channel flow surface; incidence and deviation angles at the blade leading and trailing edges; and approximations to the blade surface velocities. The output may also include input information for a blade-to-blade flow analysis program.

The program is reported in two volumes with part I as the user's manual and part II as the programmer's manual. Part I contains all the information necessary to use the program as is. It explains the equations involved and the method of solution and gives a numerical example to illustrate the use of the program. This report, part II, contains all information necessary to understand the operation of the program. It explains the overall program procedure and gives a detailed description of all the subroutines. There is also a dictionary of variable names and a complete program listing.
INTRODUCTION

The design of blades for compressors and turbines ideally requires analysis methods for unsteady, rotational, three-dimensional, viscous flow through a turbomachine. Clearly, such solutions are impossible at the present time, even on the largest and fastest computers. The usual approach at present is to analyze only steady flows and to separate inviscid solutions from viscous solutions. Three-dimensional inviscid solutions are just beginning to be contemplated for coming generations of computers. So at present, inviscid analyses usually involve a combination of several two-dimensional solutions on intersecting families of stream surfaces to obtain what is called a quasi-three-dimensional solution.

Since there are several choices of two-dimensional surfaces to analyze and many ways of combining them, there are many approaches to obtaining a quasi-three-dimensional solution. Most two-dimensional solutions are either on a blade-to-blade surface of revolution (Wu's $S_1$ surface, ref. 1) or on the meridional or mid-channel stream surface between two blades (Wu's $S_2$ surface). However, when three-dimensional effects are most important, significant information can often be obtained from a solution on a passage cross-sectional surface (normal to the flow). This is called a channel solution (see fig. 1).

Figure 1. - Two-dimensional analysis surfaces in a turbomachine.
In this report a solution to the equations of flow on the meridional surface is carried out. This solution surface is chosen when the turbomachine under consideration has significant variation in flow properties in the hub-shroud direction. A solution on the meridional surface will show this variation. The solution can be obtained either by the quasi-orthogonal method, which solves the velocity-gradient equation from hub to shroud on the meridional flow plane (ref. 2), or by a finite-difference method, which solves a finite-difference equation for stream function on the same flow plane. The quasi-orthogonal method is efficient in many cases and can obtain solutions into the transonic regime. However, there is difficulty in obtaining a solution when aspect ratios are above 1. Difficulties are also encountered with curved passages and low hub-tip ratio blades. For such cases, the most promising method is the finite-difference solution, but this solution is limited to completely subsonic flows.

Two finite-difference programs for flow on the mid-channel surface of a turbomachine have been reported in the literature (refs. 3 and 4). Since both are finite-difference methods, they are necessarily limited to subsonic flow cases. Marsh's method (ref. 3), termed the matrix throughflow method, closely follows the development given by Wu in reference 1. However, the computer program was not included in reference 3, nor is it available to the general user. Davis' program is provided in reference 4 but is limited to certain families of compressor blades and flow surfaces.

The method described in this report uses both the finite-difference and the quasi-orthogonal (velocity gradient) methods, combined in a way which takes maximum advantage of both. The finite-difference method is used to obtain a subsonic flow solution. The velocity-gradient method is then used, if necessary, to extend the range of solutions into the transonic regime.

A computer program, MERIDL, has been written to perform these calculations. This program is written for axial- or mixed-flow turbomachines, both compressors and turbines, up to approximately 45° from axial. Upstream and downstream flow conditions can vary from hub to shroud. The solution is for compressible, shock-free flow, or incompressible flow. Provision is made for an approximate correction for loss of stagnation pressure through the blade row. The blade row may be either fixed or rotating and may be twisted and leaned. The blades can have high aspect ratio and arbitrary thickness distribution.

The solution obtained by this program also provides the information necessary for a more detailed blade shape analysis on blade-to-blade surfaces (fig. 1). A useful program for this purpose is TSONIC (ref. 5). Information needed to prepare all the input for TSONIC is calculated and printed by this program.

The MERIDL program has been implemented on the NASA Lewis time-sharing IBM-TSS/360-67 computer. For the numerical example of this report, storage of vari-
bles required 60 000 words for a 21 × 41 grid of 861 points. Variable storage could be easily reduced by equivalencing of variables or by using a coarser mesh. Storage for the program code is 18 000 words. This storage could be reduced by overlay of code. Run times for the program range from 3 to 15 minutes on IBM 360-67 equipment, depending upon the mesh size used and the compressibility of the flow.

The MERIDIL program is reported in two volumes, with the user's manual presented as part I in reference 6 and the programmer's manual presented as part II in this report. Part I contains all the information necessary to use the program as is. It explains the method of solution and gives a numerical example to illustrate the use of the program. Part I includes the sections METHOD OF ANALYSIS, DESCRIPTION OF INPUT AND OUTPUT, NUMERICAL EXAMPLE, and appendixes which derive the mathematical equations used. This report, part II, contains all information necessary to understand the operation of the program. It explains the overall program procedure and gives a detailed description of all the subroutines. There is also a dictionary of variable names and a complete program listing. The appendixes explain numerical techniques used and derive certain numerical algorithms. So, part II includes the sections OVERALL PROGRAM PROCEDURE, DETAILED PROGRAM PROCEDURE, MAIN DICTIONARY, PROGRAM LISTING, and appendixes which derive the numerical methods used.

OVERALL PROGRAM PROCEDURE

This main section gives an overall view of the program calculation procedure. The next main section should be consulted for the detailed program procedure. Reference will be made to the proper section or appendix for the equations and their derivation or for the numerical techniques used.

The main program guides the overall flow of the program. All the main subroutines are called by it. Figure 2 is a flow chart for the main program.

The first step is to read and print out all the input data. This is done by the INPUT subroutine. Upstream and downstream flow conditions can be given either as a function of the streamline or as a function of radius. For program calculations, both the stream function and the radius are needed. INPUT estimates values of either stream function or radius, whichever was not given as input, based on the area distribution. These values are later adjusted with each iteration. The next step is to call INPLOT, which plots all the upstream and downstream input flow variables as well as the input blade sections from hub to tip.

The next subroutine is MESH0, which calculates the coordinates of the orthogonal
Figure 2. – Flow chart of main program.
mesh in the solution region. Details of the numerical technique are given in reference 7. After this PRECAL is called to calculate quantities which remain fixed throughout the calculations. These quantities include the s and t mesh coordinates, hub and tip wall curvatures, and leading- and trailing-edge z- and r-coordinates at horizontal mesh lines. Subroutine PRECAL also calls THETOM, THIKOM, and LOSSOM. Subroutine THETOM calculates $\partial \theta / \partial s$ and $\partial \theta / \partial t$ at the orthogonal mesh points. (All symbols are defined in appendix E.) These partials are used to calculate the blade flow angle $\beta$ and the tangential velocity $W_\theta$ after the meridional velocity $W_m$ has been calculated. Subroutine THIKOM calculates the tangential blade thickness $t_\theta$ at the orthogonal mesh points. Subroutine LOSSOM calculates the ratio of actual to ideal relative stagnation pressure downstream of the blade and then distributes the loss linearly through the blade row from leading to trailing edge. The method of making loss corrections is discussed in appendix D of part I (ref. 6). Finally, PRECAL makes corrections in mass flow, wheel speed, and whirl for the reduced-mass-flow solution if the full-mass-flow solution cannot be obtained directly (i.e., when REDFAC < 1.0).

Next MEPLOT is called to plot the meridional plane view of the blade and passage and to plot the orthogonal mesh. Then VBDRY is called to calculate the stream-function values along the upstream and downstream boundaries of the orthogonal mesh. This is done by using the velocity-gradient equation derived in appendix C of part I (ref. 6). Iteration is required to establish the correct temperature, density, and whirl to use in the velocity-gradient equation. Now INIT is called to initialize array variables as required for the first iteration. Most variables are set either to zero or to some value which will avoid division by zero later on.

At this point, everything is ready to solve the stream-function finite-difference equations. These equations are nonlinear. They are solved by an iterative procedure, with two levels of iteration. The inner iteration solves a linearized equation, and the outer iteration makes corrections to the linearized equation so that the solution converges to the solution of the original nonlinear equation. There are three subroutines called to obtain the solution to the linearized equation: COEF, SOR, and NEWRHO. Then there are four subroutines to print and plot this information and prepare for the next outer iteration: OUTPUT, INDEV, SLPLOT, and SVPLOT. These seven subroutines are repeated until convergence is obtained.

Subroutine COEF calculates the coefficients of the finite-difference equations. These coefficients are derived in appendix A. Because of the sensitivity of the calculations to the value of $\partial (rV_\theta) / \partial r$, this value is damped from iteration to iteration. Thus, only a portion of the predicted change in value is actually used. This portion is specified by the input value of DNEW.
Subroutine SOR solves the finite-difference equations for the stream function $u$ by successive overrelaxation using an optimum overrelaxation factor (ORF). This is the inner iteration. The optimum overrelaxation factor is calculated by subroutine SOR on the first iteration. Subroutine NEWRHO calculates velocity components at each mesh point by differentiating the stream function numerically along the orthogonal mesh lines. These values are used to calculate new densities at each mesh point. When whirl is not given as input, NEWRHO also makes reinitialization calls to readjust the estimated values of stream function to go with the input temperature, density, and tangential velocity. See appendix B. Subroutine NEWRHO also calculates values of $\xi$ and $\xi$ (eqs. (A1) to (A3)) at the mesh points to be used in COEF on the next iteration. And NEWRHO checks the relative change in velocity from the previous iteration at each mesh point. The maximum relative change in velocity is checked to see if the solution is converged.

Now that a solution (converged or not) has been obtained, OUTPUT is called. Subroutine OUTPUT first calculates other velocity components and flow angles at all mesh points. Then OUTPUT calculates streamline curvature and critical velocity ratio at each mesh point. Subroutine BLDVEL is called to calculate the blade surface velocities, as explained in appendix G of part I (ref. 6). Also BLDVEL calculates the average blade-to-blade density to be used in NEWRHO in the next iteration. And BLDVEL calculates $F_r$ at each point by using equation (A4). The radial vector $F_r$ is used by COEF in calculating the coefficients of the finite-difference equations. After returning from BLDVEL, OUTPUT will print out data at the orthogonal mesh points, if desired. Then, if output is desired along streamlines, the necessary interpolation will be done and data will be printed for all streamlines. Similarly, interpolation will be done and data printed for hub-tip station lines.

After OUTPUT, INDEV is called. Subroutine INDEV calculates a correction to $\partial \theta / \partial s$ for a short distance into the blade to match the mean surface within the blade to the free-stream flow angles, both upstream and downstream. The method for doing this is described in appendix F of part I (ref. 6). Subroutine INDEV also calculates and prints out incidence and deviation angles if this is requested. If desired, SLPLOT will plot the streamlines and SVPLOT will plot the mean and blade surface velocities.

At this point, the main program will start a new iteration by going back to COEF if the solution has not converged. If the solution has converged, there are two possibilities. If REDFAC is 1, the final solution has been obtained and the program is through. If there are data for another case, the program will start this case; otherwise the program is stopped. If REDFAC is less than 1, the final approximate full-mass-flow solution will be calculated by TVELCY. First, the mass flow, rotational speed, and inlet and output whirl are restored to their full values. This requires reinitialization calls.
of LAMDAF and RVTHTA for inlet and outlet whirl. Then TVELCY calculates \( \partial W_m / \partial m \) and \( \partial W_\theta / \partial m \) for use in the velocity-gradient equation. These quantities are first calculated from the reduced-mass-flow solution and then are adjusted by dividing by REDFAC. Now the velocity-gradient equation (derived in appendix C of part I (ref. 6)) is solved along each vertical mesh line. Iteration is required to establish the correct temperature, density, and whirl to use in the velocity-gradient equation. When TVELCY is through, TOUTPT is called. Subroutine TOUTPT is an alternate entry point for OUTPUT. The only difference is that the flow angles are considered to be known, and the velocity components are calculated from the velocity magnitude and the known flow angles. Then the same sequence of INDEV, SLPLOT, and SVPLOT is called as for the finite-difference solution. Normally, only the smaller (subsonic) of two possible solutions is obtained by TVELCY (part I, appendix C); but if desired, both the larger ("supersonic") and smaller solutions can be obtained. If both solutions are desired, TVELCY, TOUTPT, INDEV, SLPLOT, and SVPLOT are called again. This completes the program. If there are data for another case, the program will start on this case: otherwise the program is stopped.

**DETAILED PROGRAM PROCEDURE**

This main section gives the detailed program procedure for all the subroutines. The previous main section should be consulted for an overall view of the program calculation procedure.

Most of the subroutines in MERIDL use the same set of variables. These variables are all defined in the MAIN DICTIONARY. All subroutines are described prior to the main dictionary. First, the main subroutines and other subroutines which use the main dictionary are described, and then the remaining subroutines with special dictionaries are described.

The calling relation of all subroutines is shown in figure 3. Note that figure 3 is not a flow chart. A tabulation of all subroutines called and all COMMON blocks for each subroutine is given in table I.

The first subsections presented here describe the general aspects of the programs, including storage requirements, conventions used, and description of labeled COMMON blocks. They are followed by a detailed description of the subroutines.
Figure 3. - Calling relation of subroutines. Called subroutines are always below the calling subroutine. (This is not a flow chart.)
<table>
<thead>
<tr>
<th>Subroutine name</th>
<th>COMMON blocks (a)</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
<th>Subroutine name</th>
<th>COMMON blocks (a)</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERIDL</td>
<td>/---/</td>
<td>INPUT INPLOT</td>
<td>None - main</td>
<td>MEPLT</td>
<td>/INPUTT/</td>
<td>PTBDRY</td>
<td>MERIDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MERISNL MICSPL</td>
<td>program</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRECAL MERIDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEPLT VDBDRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIST COEF SOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NEWSRH OUTPUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDEVL SLPLT SVPLOT TVELCY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOUTPT (OUTPUT)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPUT</td>
<td>/---/</td>
<td>SPLINT MERIDL</td>
<td></td>
<td>CONTIN</td>
<td>None</td>
<td>None</td>
<td>CONTIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPLOT</td>
<td>/INPUTT/</td>
<td>Microfilm plot</td>
<td>MERIDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>subroutines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPLINT SPLINE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESHO</td>
<td>/INPUTT/</td>
<td>CROSCD MERIDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPINT SPINE ROOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CROSCD</td>
<td>/INPUTT/</td>
<td>None MESHO ROOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRECAL</td>
<td>/INPUTT/</td>
<td>LAMNT (LAMDAF)</td>
<td>MERIDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RVTNIT (RVTHTA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIPNIT (TIPF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RHINIT (RHOIF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RHONIT (RHOOPF)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPLINE INSECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPLINE SPLINT TETOM TETOM TETOM TETOM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOSSOM LOSSOM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THEOM</td>
<td>/INPUTT/</td>
<td>SPLINT SPLINE LININT</td>
<td>PRECAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THIKOM</td>
<td>/INPUTT/</td>
<td>SPLINE LININT</td>
<td>PRECAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOSSOM</td>
<td>/---/</td>
<td>TIPF LAMDAF TOPF RHOIF SPLINT</td>
<td>PRECAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OUTPUT (entry point - TOUTPT)</td>
<td>/---/</td>
<td></td>
<td>MERIDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SLOPES TIPF</td>
<td>MERIDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LAMDAF RVTHTA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TIPNIT (TIPF)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RHINIT (RHOIF)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RHONIT (RHOOPF)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LOSSOM</td>
<td></td>
</tr>
</tbody>
</table>

--- denotes an unlabeled COMMON block.
<table>
<thead>
<tr>
<th>Subroutine name or entry point</th>
<th>COMMON blocks</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
<th>Subroutine name or entry point</th>
<th>COMMON blocks</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLVEL</td>
<td>/---/</td>
<td>SLOPES</td>
<td>OUTPUT</td>
<td>RHOOPF</td>
<td>/INPUT/</td>
<td>None</td>
<td>VBDRY TVELCY</td>
</tr>
<tr>
<td>IUTE</td>
<td>/INPUT/</td>
<td>LAMDAF</td>
<td>TIPF RHOOPF</td>
<td>RHONT (entry point for RHOOPF)</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>INDEV</td>
<td>/INPUT/</td>
<td>SPLINT</td>
<td>OUTPUT</td>
<td>RHONT (see entry RHONT)</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>SPLLOT</td>
<td>/---/</td>
<td>Microfilm plot</td>
<td>MERIDL</td>
<td>RVTNIT (entry point for RVTNIT)</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>SPLOT</td>
<td>/---/</td>
<td>Microfilm plot</td>
<td>MERIDL</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>TVELCY</td>
<td>/---/</td>
<td>LAMNT (LAMDAF)</td>
<td>MERIDL</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>TIPF (see entry TIPNIT)</td>
<td>/INPUT/</td>
<td>None</td>
<td>LAMNT</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>TIPNIT (entry point for TIPF)</td>
<td>/INPUT/</td>
<td>SPLINE</td>
<td>PRECAL NEWRHO</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>RHOOPF (see entry RHONT)</td>
<td>/INPUT/</td>
<td>None</td>
<td>LAMNT</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>RHONT (entry point for RHOOPF)</td>
<td>/INPUT/</td>
<td>SPLINE</td>
<td>PRECAL NEWRHO</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>LAMDAF (see entry LAMNIT)</td>
<td>/---/</td>
<td>None</td>
<td>LAMNT</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>LAMNIT (entry point for LAMDAF)</td>
<td>/---/</td>
<td>MANMT</td>
<td>PRECAL NEWRHO</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
<tr>
<td>TOPF</td>
<td>INPUT CALCON</td>
<td>TIPF LAMDAF RVTN</td>
<td>TVELCY</td>
<td>None</td>
<td>/INPUT/</td>
<td>SPLINE TOPF</td>
<td>PRECAL NEWRHO</td>
</tr>
</tbody>
</table>

--- denotes an unlabeled COMMON block.
STORAGE REQUIREMENTS

The MERIDL program has been implemented on the NASA Lewis time-sharing IBM-TSS/360-67 computer. Storage for the program code is approximately 18 000 words. For the numerical example of part I (ref. 6), storage of variables required approximately 60 000 words for a $21 \times 41$ grid of 861 points. As dimensioned for a $100 \times 101$ grid, storage of variables would require about 680 000 words. The user can reduce the storage requirements for variables, as desired, by changing the dimensions. The main dictionary indicates how each variable should be dimensioned to reduce the storage required. This is indicated by reference to certain input variables, such as MM, MHT, NHUB, NTIP, NBLPL, NPPP, and so forth. The variables with the most significant effect on storage requirements are MM and MHT.

As an example, consider the two-dimensional array ALPHA. This variable is in the /VARCOM/ COMMON block and is dimensioned ALPHA (100,101) in the program listing. In the main dictionary, it is listed as ALPHA (MM, MHTP1). Suppose that the maximum desired value for MM is 60 and that for MHT it is 40. Since MHTP1 is MHT + 1, the maximum value for MHTP1 would be 41. Then ALPHA should be dimensioned ALPHA (60,41).

Similarly, all other dimensioned variables should have their dimension changed as required. Most dimensioned variables are in COMMON blocks, but there are a few which are dimensioned locally only. In addition, the calls to LININT must be changed to reflect any changes in the dimensions of the first two LININT arguments.

CONVENTIONS USED IN PROGRAM

For convenience, a number of conventions are used in naming variables and assigning subscripts.

In addition to the basic orthogonal mesh, there are four special mesh schemes used, as illustrated in figure 4. For each mesh, different conventions are used to indicate mesh position. The subscripts I and J are used to denote orthogonal mesh position. The I is used to denote the vertical mesh number, and the J is used to denote the horizontal mesh line number. The subscripts IS and JS are used in a similar manner to denote streamline mesh points, and IL and JL the station-line mesh points. Likewise, IN and JN denote points on the input blade sections, and KN and JN denote points on the alter-
Figure 4. - Five meshs used in MERIDL.
nate blade mesh located at 10-percent-chord intervals in the THETOM subroutine. Note
that I and IS take on the same values, as do JS and JL.

In variable names, I or IN indicates the inlet (upstream of blade) and O or OUT
indicates the outlet. Variables ending with OM are generally variables defined on the
orthogonal mesh.

Velocity components on the orthogonal mesh usually have SUB in the name, such as
WSUBZ for $W_z$. Velocity components along streamlines end in SL (WZSL), while
velocity components on station lines end in ST (WZST). The letters H or HUB in a var-
iable name indicate the hub, and T or TIP the tip. LE is used for leading edge and TE
for trailing edge. The letters TH indicate a variable in the $\theta$-direction, SURF a varia-
ble on a blade surface, and BL a variable in the blade region. In a variable name,
TEM indicates a temporary variable; P is used to indicate a prime superscript, and PP
double prime; D is used for derivative. Usually, several conventions are combined in
each variable. For example, TIP is used for $T'_1$, TPPTIP for $T''/T'_1$, and DPDR is
used for $\partial p/\partial r$.

All subroutines used for plotting have PLOT in the name. Variables used for plot-
ting have PLT in the name.

**LABELED COMMON BLOCKS**

Most variables which are used in more than one subroutine are placed in labeled
COMMON blocks. A brief description of each labeled block is given. The same vari-
able names are used in different subroutines for every variable in a COMMON block.
The labeled COMMON blocks are as follows:

/INPUTT/ is used for all input quantities.
/CALCON/ is used for calculated constants which are initially calculated and
are not changed later.
/VARCOM/ is used for all orthogonal mesh point arrays which are changed in
each iteration.
/SLCOM/ is used for output data along streamlines.
/STACOM/ is used for output data along station lines.
/INDCOM/ is used for quantities calculated by THETOM to be used by INDEV.
/PLTCOM/ is used to plot data for hub, shroud, and blade leading and trailing
edges.
/CROSCM/ is used to store quantities required by CROSCD.
/INTITL/ is used to store the input title for use by INPLOT.

Table I shows which COMMON blocks are needed in each subroutine.
MAIN PROGRAM

The program is segmented into several main subroutines called by the main program, as indicated in figure 3. The subroutines are called in sequence, except for the outer iteration and a switch to obtain a supersonic final solution. The outer iteration is a loop consisting of calls to COEF, SOR, NEWRHO, OUTPUT, INDEV, SLPLOT, and VPLOT. This calling sequence and the outer iteration loop are shown more clearly in the flow chart for the main program, given in figure 2. Flow charts for some of the subroutines are also given with the subroutine descriptions.

SUBROUTINES

Subroutine INPUT

Subroutine INPUT reads and prints all input data cards and initializes some variables for use later in the program.

All input cards are first read and printed on the output listing in the same form and order in which they are given. All array bounds are then checked to see if they are within limits, and some miscellaneous constants are initialized. Finally, estimates are made of various required upstream and downstream flow conditions which were not given as input because other input options were used.

Subroutine INPLOT

Subroutine INPLOT makes microfilm plots of a portion of the given input data. By checking these plots, the user can see if his input data have been given correctly and smoothly. It is important that the plotted data be smooth, since spline curve fits of these data are used extensively by the program.

Two main sections of input are plotted: the upstream and downstream flow conditions, and the input blade sections from hub to shroud. A separate plot is made of each of the three given distributions of upstream flow variables and two distributions of downstream flow variables from hub to shroud. On each plot, one data point is plotted at every 1 percent of stream function or radius from hub to shroud. The NIN and NOUT input points are also marked. Each input blade section is then plotted, using only the input points for plotting and marking. After each individual blade section is plotted, a multiple plot is made of all sections together. Examples of all the plots are given in figures (a) to (g) of part I (ref. 6).
Subroutine INPLOT and the other plot routines, MEPLLOT, SLPLLOT, and SVPLLOT, all rely heavily on the NASA Lewis in-house microfilm plotting package described in reference 8. These four routines as well as PTBDRY, which is called by MEPLLOT and SLPLLOT, are self-contained and can be easily removed from MERIDL without disturbing the remainder of the calculations.

Subroutine MESHO

Subroutine MESHO calculates the coordinates of an orthogonal mesh covering the solution region from upstream to downstream of the blade row and from hub to shroud. Subroutine MESHO makes use of four other subroutines - ROOT, CROS CD, SPLINE, and SPLINT. A flow chart for MESHO is given in figure 5. The method used for generating the mesh is explained thoroughly in reference 7.

Subroutine MESHO begins with input geometry describing the hub and shroud of the flow passage and the numbers of mesh points desired in the horizontal and vertical directions. First, MESHO calculates the horizontal, or streamwise, orthogonals. It does this by extending lines vertically from each of the input points on the hub to the shroud. Each of these lines is then divided into equal increments, the number depending upon the number of streamwise orthogonals. Streamwise spline curves are fit through the resulting points to give the horizontal orthogonals shown in figure 6.

Vertical orthogonal lines are then constructed one at a time, moving from left to right between each pair of adjacent horizontal orthogonals, proceeding from hub to shroud, as shown in figure 7. The procedure for calculating these lines, shown in figure 8, is analogous to a technique for solving ordinary differential equations known as the improved Euler method or Heun’s method (ref. 9). Beginning at a known orthogonal mesh point on the lower orthogonal, a normal is constructed (line \( \text{1} \) in fig. 8) to the upper orthogonal. Then the intersection coordinates of this line with the upper orthogonal and the slope of the upper orthogonal at the intersection are calculated. ROOT and CROS CD are used in this process. Line \( \text{2} \) in figure 8 is then constructed in such a way that it is perpendicular to the tangent to the upper orthogonal at the intersection point and passes through the original starting point on the lower orthogonal. The coordinates of the intersections of both lines \( \text{1} \) and \( \text{2} \) are now known on the upper orthogonal. The desired new orthogonal mesh point is the average of these two sets of coordinates.

This process of constructing vertical orthogonal links is continued until the shroud is reached by all vertical orthogonals. This completes the generation of the orthogonal mesh.

Notice in MESHO that the locations of the upstream and downstream boundaries of the orthogonal mesh at the hub are fixed by the inputs ZOMIN and ZOMOUT (fig. 7). The
Extend radial lines from hub; calculate r-coordinate array, RRAD

Calculate z-coordinates of orthogonal mesh at the hub, ZOM

Begin construction of vertical orthogonals; proceed row by row; \( i = 2 \)

Calculate r-coordinates, slopes, and angles of mesh, ROM, SLOM, CPHI, SPHI, for \( j = 1 \) streamwise orthogonal

Calculate second derivatives, SDRIV, on present orthogonal

Begin construction of vertical orthogonal links between two streamwise orthogonals; \( J = J + 1 \)

Locate position of vertical orthogonal link among RRAD points

Using ROOT, calculate z- and r-sing CROSCD, calculate coordinates, Z1, R1, and slope, SL1, then z- and r-slope, SL, of intersection coordinates of intersection of normal to lower orthogonal with upper orthogonal

Using CROSCD, calculate slope, SL1, then z- and r-coordinates of intersection of normal to lower orthogonal with upper orthogonal

Calculate z- and r-coordinates, Z2, R2, of intersection of normal from upper orthogonal to mesh point on lower orthogonal

Calculate average z-coordinate of two intersection points on upper orthogonal, ZOM(I,J)

No

\( i = i + 1 \)

No

\( J = J + 1 \)

No

\( J > MHTP1 \)

Yes

Calculate r-coordinates and angles of orthogonal mesh, ROM, CPHI, SPHI, for final streamwise orthogonal

Return

Figure 5. - Flow chart for MESHO.
Horizontal orthogonals - Radial lines divided into equal increments

Input points on hub (ZHub, RHUB)

Figure 6. - "Horizontal" orthogonals obtained by spline curve fitting.

Figure 7. - Process for generating "vertical" orthogonal links.
locations of these boundaries at the tip, however, cannot be given ahead of time and are totally dependent upon the orthogonal mesh generation procedure.

Axial distance between vertical orthogonal origins at the hub is determined by the number of mesh lines requested in the following three regions: MBI mesh lines upstream of the blade from ZOMIN to ZOMBI; MBO - MBI mesh lines from ZOMBI to ZOMBO; and MM - MBO mesh lines downstream of the blade from ZOMBO to ZOMOUT (fig. 7). The number of horizontal orthogonals is MHT + 1, which is the same in all three regions.

Subroutine CROSCD

The ROOT subroutine (p. 52) requires the calling of a special function or subroutine. In MERIDL, that routine is CROSCD. It is called by ROOT to calculate for a given z-coordinate the difference in r-coordinates of line 1 and the upper horizontal orthogonal in figure 8. (ROOT finds the z-coordinate where this difference shrinks to zero, that is, the intersection of the straight line 1 and the horizontal orthogonal curve.)

The input argument for CROSCD is

Z value of z-coordinate in ROOT

The following two values are given as output:

RMR difference in r-coordinates of straight line and curve

SL1 slope of horizontal orthogonal at z

CROSCD uses the equations of a cubic spline curve to interpolate and calculate r as
a function of $z$ on the horizontal orthogonal. The spline curve information is transmitted to CROSCD from MESHO in the /CROSCM/ COMMON.

Subroutine PRECAL

Subroutine PRECAL calculates many of the fixed constants which will be needed by the subroutines in the outer iterative loop of MERIDL. Figure 9 gives a flow chart for PRECAL.

First, PRECAL initializes the subroutines for calculating upstream and downstream flow conditions. To do this it calls LAMDAF, RVTHTA, TIPF, RHOIPF, and RHOOPF, entering at the special entry points of these routines used for initialization.

The array of blade-to-blade spacing $B$ (the BTH array) is then initialized to the blade pitch (in radians) at every point on the solution mesh. This array is modified in the blade region later in PRECAL when THIKOM and LOSSOM are called.

In the cases where output streamline values (FLFR array) were not read in (NSL = 0), PRECAL assigns eleven (11) values to FLFR from 0 to 1.0, in increments of 0.1. Also, if the given endpoints of FLFR do not equal 0 and 1.0, PRECAL adds these values as endpoints.

Then, PRECAL uses the $z$- and $r$-coordinates of the orthogonal mesh (ZOM and ROM), calculated in MESHO, to calculate the $s$ and $t$ arrays (SOM and TOM) on the orthogonal mesh. Adjacent points are linked with straight line segments in this calculation of $s$ and $t$, but the correction between arc length and chord length is not significant for adjacent points.

The curvatures of the hub and shroud profiles are then calculated where these profiles are intersected by the upstream and downstream boundaries of the orthogonal mesh. These curvatures are later required in the VBDRY subroutine.

The $z$- and $r$-coordinate arrays (ZLE, RLE and ZTE, RTE) are then set up at points which define the leading and trailing edges of the blade. These values are the first and last values for each blade plane from the ZBL and RBL blade-coordinate input arrays. The intersections of these leading and trailing edges with the hub and shroud are also calculated with INRSCT calls.

Various quantities are then calculated on the orthogonal mesh at or near the leading and trailing edges of the blade. With INRSCT calls, the $z$- and $r$-coordinates of intersections of horizontal mesh lines with the blade edges are calculated. Vertical mesh line numbers (ILE and ITE) of mesh points which lie just within the blade leading and trailing edges are then calculated by comparing the $z$-coordinates of mesh points along the orthogonals with the $z$-coordinates of intersections of the orthogonals with the blade edges. The $s$- and $t$-coordinates are then calculated for the points where the horizontal mesh lines cross the blade edges.
Initialize subroutines LAMDAF, RHOOTA, TIPF, and RHDIPF by calculating splines of inputs against stream function.

- If given as input?
  - Yes, calculate E from mass.
  - No, initialize subroutine RHOOPF.

  - Initialize EHTH to pitch.
  - There output streamline locations specified (FLFR)?
    - Yes, set EHTH and points = 0.0, 1.0.
    - No, initialize 11 streamline locations.

- Calculate s and t coordinates of orthogonal mesh, SOM and TCM.
- Calculate curvature of hub and shroud at mesh boundaries, CURVI, CURVIH, CURVII, CURVIIH.
- Store from input blade section arrays, ZBL and RBL, into leading- and trailing-edge arrays, ZLE, RLE, ZTE, RTE.
- Calculate intersections of blade edges with hub and shroud.
  - Calculate z, r, s, and 0 where orthogonal mesh crosses blade edges; calculate ZLE and ZIE.
  - THEROM: Calculate 809s and 809t.
    - THEROM: Correct pitch, BTH, for blade thickness.
    - LOSSOM: Correct pitch, BTH, for total pressure loss.
      - REDFAC < 1.0?
        - Yes, return.
        - Yes, reduce w, u, a, r, v, for reduced-flow solution.
  - Reinitialize subroutines LAMDAF and RIVTHA for reduced-flow solution.
  - Return.

Figure 9. Flow chart for PRECAL.
Then PRECAL calls three other subroutines, THETOM, THIKOM, and LOSSOM. The THETOM routine calculates $\partial \theta / \partial s$ and $\partial \theta / \partial t$ at the orthogonal mesh points. Subroutine THIKOM makes corrections to the BTH array to account for blade thickness, and LOSSOM calculates the relative total pressure loss at the downstream boundary of the orthogonal mesh. This loss is distributed linearly through the blade row by making an additional correction to the BTH array.

Finally, in PRECAL, corrections are made to some upstream and downstream input arrays and corresponding boundary conditions in the case where a reduced-mass-flow solution is to be obtained (REDFAC < 1.0). The wheel speed, mass flow, whirl, and tangential velocity are all reduced by REDFAC; and the upstream and downstream boundary conditions of whirl are reinitialized by LAMDAF and RVTHTA calls.

Subroutine THETOM

Subroutine THETOM calculates the gradients $\partial \theta / \partial s$ and $\partial \theta / \partial t$ at the orthogonal mesh points which lie within the leading and trailing edges of the blade. This process is thoroughly described in appendix C.

Theta coordinates of the mean blade surface (THBL) are given at the input blade section points (ZBL, RBL). Gradients of the $\theta$-coordinate are required in the $s$- and $t$-directions at the orthogonal mesh points within the blade for use by the NEWRHO subroutine.

Subroutine THETOM makes use of the technique of defining an alternate mesh which is entirely contained within the blade on which $\partial \theta / \partial z$ and $\partial \theta / \partial r$ are obtained. By interpolation, $\partial \theta / \partial z$ and $\partial \theta / \partial r$ are then obtained at the required orthogonal mesh points. Finally, $\partial \theta / \partial s$ and $\partial \theta / \partial t$ are calculated from $\partial \theta / \partial z$ and $\partial \theta / \partial r$ at these points.

Subroutine THIKOM

Subroutine THIKOM calculates the blade thickness in the $\theta$-direction at the points of the orthogonal mesh which lie within the blade edges. (Input blade thicknesses are not given at the orthogonal mesh points, nor are they given in the $\theta$-direction. They are given normal to the blade mean camber line along each input blade section.)

THIKOM first calculates the $s'$-coordinate and then the angle $\kappa$ between the mean camber line and the $s'$-coordinate direction, as shown in figure 10. (The $s'$-coordinate corresponds to the input blade section direction.) With these angles, approximate thicknesses in the tangential direction $t_\theta$ are calculated from thicknesses normal to the meanline $t_n$ by the equation...
This calculation is subject to error for highly cambered or highly staggered blade sections but is adequate here since \( t_{\theta} \) is only used as a blockage correction to the BTH array.

After \( t_{\theta} \) is obtained at the input points, LININT is called to interpolate and obtain it at the orthogonal mesh points. Then it is subtracted from the BTH array.

\[
t_{\theta} = \frac{t_n}{\cos \kappa}
\]

Figure 10. - Calculation of thickness in tangential direction.

Subroutine LOSSOM

Subroutine LOSSOM calculates the downstream relative total pressure loss and distributes it upstream through the blade row as an area correction. This correction is made to the BTH array. The loss is calculated as 1 minus the ratio of actual to ideal relative total pressure along the hub-shroud input line downstream of the blade.

\[
\text{Loss} = 1 - \frac{p''_0}{\left(p''_0\right)_{\text{ideal}}}
\]

In one input option, \( p'_1, T'_1, \) and \( p'_0 \) are given and \( T'_0 \) is then calculated from Euler's equation. Then using the relations
\[
\frac{p''_o}{p'_o} = \frac{T''_o}{T'_o}^{\gamma/(\gamma-1)}
\]

and

\[
\frac{(p''_o)_{\text{ideal}}}{p'_i} = \frac{T''_o}{T'_i}^{\gamma/(\gamma-1)}
\]

we form the ratio of actual to ideal relative total pressure

\[
\frac{p''_o}{(p''_o)_{\text{ideal}}} = \frac{p'_o}{p'_i} \left(\frac{T'_i}{T'_o}\right)^{\gamma/(\gamma-1)}
\]

There is an alternate input option where loss is given as input. In that case, relative total pressure ratio is calculated as

\[
\frac{p''_o}{(p''_o)_{\text{ideal}}} = 1 - \text{Loss}
\]

Subroutine LOSSOM then estimates the stream function, based on area, at points on the downstream boundary of the orthogonal mesh. Subroutine SPLINT is then called to interpolate the values of relative total pressure ratio at these same downstream boundary mesh points. From the interpolated values of pressure ratio, loss is computed by using the preceding equations. It is assumed that the horizontal mesh lines are close approximations to the actual streamlines. Thus, the loss is distributed along horizontal mesh lines. Along any mesh line, loss is assumed to be constant in the region downstream of the blade trailing edge and equal to the downstream boundary value. Between the blade leading and trailing edges, loss is distributed linearly from zero at the leading edge to full value at the trailing edge. It is assumed that no loss occurs upstream of the blade. Loss is included in the stream-function solution by reducing the value of \( B \) as follows:

\[
B_{\text{net}} = B(1 - \text{Loss})
\]

where \( B_{\text{net}} \) is the final value stored in the BTH array.
Subroutine MEPLLOT

Subroutine MEPLLOT makes two microfilm plots of the blade in the meridional plane. The first plot shows the input hub and shroud geometry and the blade leading and trailing edges. The edges are obtained from the ZBL and RBL input arrays. The second plot shows the same hub, shroud, and blade but with the generated orthogonal mesh superimposed on the solution region. Examples of these plots are given in figures 16(h) and (i) of part I (ref. 6).

Prior to making these two plots, MEPLLOT calls PTBDRY to obtain the boundary points to be plotted on the hub, the shroud, and the blade edges and to scale the plots.

Subroutine PTBDRY

Subroutine PTBDRY obtains coordinates used in plotting the hub, the shroud, and the blade edges by the two routines MEPLLOT and SLPLLOT. Using SLPLLOT calls, PTBDRY interpolates on the input arrays ZHUB, RHUB and ZTIP, RTIP to obtain 100 plotting points on both hub and shroud. The same is done with the blade leading and trailing edges. After the plot points are obtained, PTBDRY searches the range of values to be plotted for the maximum in both X- and Y-directions and adjusts the range of plotted points so that it is the same in both directions. The computed information is stored in the /PLTCOM/ COMMON.

Subroutine VBDRY

Subroutine VBDRY calculates the value of the stream function along the upstream and downstream boundaries of the mesh region. The calculation is based on the velocity-gradient equation (C9) of part I (ref. 6).

There are four arguments for VBDRY so that the same coding can be used for either the upstream or downstream boundary. These arguments are LOC, TIPF, RHOIPF, and LAMDAF. The argument LOC is the value of I for the desired boundary, that is, LOC = 1 on the upstream boundary and LOC = MM on the downstream boundary. The other three arguments are function subroutines. That is, TIPF, RHOIPF, and LAMDAF will refer to the subroutines of the same name at the upstream boundary but will refer to TOPF, RHOOPF, and RVTHTA, respectively, at the downstream boundary. Figure 11 is a flow chart for VBDRY.

The first step is to set CURVH and CURVT to the value of the meridional curvature at the hub and tip. For this, values previously calculated in PRECAL are used. To
start the iterative procedure, a reasonable estimate of the hub relative velocity is needed (WHUB). This estimate is based on a one-dimensional calculation. Before any coefficients for the velocity-gradient equation can be calculated, values for whirl and free-stream absolute stagnation temperature and density are needed. These are all functions of stream function. An initial value of stream function is estimated based on area distribution. Values for whirl and free-stream absolute stagnation temperature and density are then obtained from subroutines LAMDAF, TIPF, and RHOIPF and are stored in arrays.
Now the coefficients of the velocity-gradient equation (C9) of part I (ref. 6) can be calculated. Either equation (C10) or (C11) of part I will be used, whichever is appropriate. The curvature, 1/\(r_c\), is assumed to vary linearly along the boundary. It is assumed that the boundary is normal to the streamlines so that \(\alpha = \varphi\). The quantity \(\cos \varphi\) has been previously calculated by MESHO.

After the coefficients are calculated, the velocity-gradient equation is solved numerically. WHUB is the initial value of \(W\) on the hub. The first iteration will use the value of WHUB calculated previously by VBDRY. Later iterations will use estimated values calculated by CONTIN. Once WHUB is specified, the numerical solution to the velocity-gradient equation is calculated by the Heun method (ref. 9). The equations used in the Heun method for this case are

\[
\begin{align*}
W_{j+1}^* &= W_j + (dW)_j \quad \text{first estimate of } W_{j+1} \\
W_{j+1}^{**} &= W_j + (dW)^*_j \quad \text{second estimate of } W_{j+1} \\
W_{j+1} &= \frac{W_{j+1}^* + W_{j+1}^{**}}{2} \quad \text{average of two estimates of } W_{j+1}
\end{align*}
\]

where \((dW)_j\) (eq. (C9) of part I) is evaluated at \(t_j\) and \(W_j\), and where \((dW)^*_{j+1}\) is evaluated at \(t_{j+1}\) and \(W^*_{j+1}\). At the same time the solution of the velocity-gradient equation is being calculated, the mass flow integration is also being calculated by trapezoidal integration:

\[
w = \int_0^{t_{\text{tip}}} \rho W \cos \beta r B \, dt
\]

The equations used in calculating the integrand are (B13), (C5), (C6), (C7), and (D4) of part I (ref. 6). At the end of the DO loop at statement 80, the integrated mass flow, UOM (LOC, MHTPI), has been calculated for the specified value of WHUB. This value is checked to see if it is within the tolerance to MSFL. If not, CONTIN is called to provide the next estimate for WHUB. See CONTIN for a description of the procedure for finding the correct value for WHUB. Provision is made to adjust WHUB to avoid problems in calculating either \(\beta\) (if WHUB is too small) or \(T/T_i\) (if WHUB is too large). (See eqs. (C6) or (B13) and (D4) of part I.) After a few iterations, usually four or five and not more than 100, a solution will be found that satisfies continuity. This completes the inner iteration of this subroutine. Then the values of the absolute stagnation tem-
perature $T_i'$ or $T_0'$, the absolute stagnation density $\rho_i'$ or $\rho_0'$, and the whirl $\lambda$ or $\left( rV_\theta \right)_0$ are recalculated from the new stream-function values. If there is a significant change in any of these values, the solution will be repeated ($\text{REPEAT} = .\text{TRUE}.$).

When a final acceptable solution is found, the program returns to the main program. If the passage is choked or if an acceptable solution cannot be found in 1000 total iterations, the program will print an error message and stop. See the section Error Messages in part I (ref. 6) for suggestions on what to do when an error message is encountered.

Subroutine INIT

This subroutine initializes certain arrays in /VARCOM/. This is necessary to start the outer iteration running from COEF to SVPLOT. For the initial iteration, it is assumed that $\rho = \rho_i'$ throughout the passage. All other values are set to zero, except for $W_s$, $W_t$, and $W_z$, which are set to values which will avoid division by zero.

Subroutine COEF

Subroutine COEF calculates the coefficients $a_1$, $a_2$, $a_3$, and $a_4$ and the constants $k_0$ for the finite-difference equations. The finite-difference equation is (A5) or (A7). The coefficients are calculated by the procedure of equation (A8), and the constants are calculated by equation (A9). Within the blade row, the value of the constant $k_0$ depends on $\partial(rV_\theta)/\partial r$. This gradient tends to be unstable with iteration, so that usually damping is required between iterations. The damping rate is controlled by the input variable DNEW. Suggestions for choosing proper values for DNEW are given in the INPUT section of part I (ref. 6). For every outer iteration, the maximum and minimum values of $\partial(rV_\theta)/\partial r$ and the maximum predicted change in $\partial(rV_\theta)/\partial r$ are calculated and printed. When it is indicated by the value of IDEBUG, the coefficients $a_i$ and the constants $k_0$ will be printed.

Subroutine SOR

Subroutine SOR solves the finite-difference equations (A5) by the method of over-relaxation (ref. 10). Equation (A5) holds at every interior point of the orthogonal mesh where the value of $u$ is initially unknown. Thus, if there are $n$ interior points, we have $n$ equations with $n$ unknowns. Equation (A5) is nonlinear but can be linearized by using values from the previous outer iteration for the nonlinear terms or factors.
SOR solves only the linearized equations.

The overrelaxation iteration is the inner iteration; it is optimized by using an optimum overrelaxation factor (ORF). The calculation of ORF is done only the first time that SOR is called. The optimum value for the overrelaxation factor $\Omega$ is estimated by using equations (B3) and (B1) of reference 11. At each interior point, $u_m^{m+1}$ is calculated from the values of $u$ at the neighboring points by

$$u_m^{m+1} = \sum_{i=1}^{4} a_i u_i$$

where each $u_i$ is the most recently calculated value for the point. To start, $u_0^0 = 1$ at the interior points and $u_0^0 = 0$ at the boundary points. The maximum (LMAX) and minimum (LMIN) values over all the interior mesh points of the ratio $u_m^{m+1}/u_m^m$ are calculated for $m = 1, 2, 3, \ldots$ until the LMAX and LMIN ratios are close to each other. Then the optimum overrelaxation factor (ORF) is calculated by

$$ORF = 2 / \left(1 + \sqrt{1 - \text{LMAX}}\right).$$

The theory for calculating ORF is derived in reference 10.

With an optimum value for the overrelaxation factor $\Omega$, the solution to equation (A5) is calculated by overrelaxation by

$$u_m^{m+1} = u_m^m + \Omega \left( \sum_{i=1}^{4} a_i u_i + k_0 - u_m^m \right)$$

where each $u_i$ is the most recently calculated value at an interior point or is a boundary value. During each iteration, the maximum change of the stream function is calculated. When this maximum change is reduced below $10^{-5}$, the iteration is stopped, and the current estimate of the stream function is accepted as the solution.

Subroutine NEWRHO

Subroutine NEWRHO calculates the velocity magnitude and components and the density at each point of the orthogonal mesh. Figure 12 is a flow chart for NEWRHO.

Normally, the upstream and downstream flow conditions, including whirl, are given as a function of the stream function. However, this information may be given as a function of position from hub to tip. In this case, an initial estimate of streamline position is made in PRECAL. Then adjustments are made in each iteration. This is done in...
NEWRHO by reinitializing the subroutines for calculating upstream and downstream flow conditions (LAMDAF, RVHTA, TIPF, RHOIPF, and RHOOPF). An explanation of how upstream and downstream flow conditions are matched to the stream function solution is given in appendix B.

The main function of NEWRHO is to calculate the partial derivatives of the stream function in the s- and t-directions. These partials are used to calculate the velocity components. These components, together with either the blade shape or the specified whirl, determine the relative velocity magnitude. With the relative velocity known, the density can be calculated. Subroutine NEWRHO calculates $\xi$ and $\zeta$ for the next iteration.

The first major loop in NEWRHO calculates $\partial I/\partial s$ and $\partial p''/\partial s$. This is done by first calculating $I$ and $p''$ along the horizontal mesh lines. The actual relative stagnation pressure $p''$ is calculated by
\[ p'' = p'_1 R T'_1 \left( \frac{T''}{T'_1} \right)^{\gamma / (\gamma - 1)} \left( 1 - \frac{p'_{\text{ideal}} - p''}{p'_{\text{ideal}}} \right) \]  

(2)

where

\[ \frac{T''}{T'_1} = 1 - \frac{2\omega_t - (\omega r)^2}{2c_p T'_1} \]  

(3)

Equation (3) is the same as equation (B13) of part I with \( W = 0 \). The rothalpy \( I \) is calculated from equation (B7) of part I (ref. 6). Then, \( \partial I / \partial s \) and \( \partial p'' / \partial s \) are calculated by calling the subroutine SLOPES.

The next loop calculates \( W_t \). First, SPLINE is called to calculate \( \partial u / \partial s \) along horizontal mesh lines. Then \( W_t \) is calculated by equation (G11) of part I.

The final major loop calculates partial derivatives in the \( t \)-direction and then calculates \( W_s, W_\theta, V_\theta, W, \rho, \xi, \) and \( \xi \) at every mesh point. The first inner loop calculates \( T''/T'_1 \) and \( p'' \) by equations (3) and (2) along vertical mesh lines. The values of the \( t \)-coordinate and stream function \( u \) are also stored in temporary arrays. Then SPLINE is called to calculate \( \partial u / \partial t \), and SLOPES is called twice to calculate \( \partial I / \partial t \) and \( \partial p'' / \partial t \). The second inner loop performs the remaining calculations. Equation (G10) of part I is used to calculate \( W_s \). Within the blade \( W_\theta \) can be calculated from \( W_s, W_t, \partial \theta / \partial s, \) and \( \partial \theta / \partial t \). Since

\[ W_\theta = W_m \tan \beta \]

\[ \tan \beta = r \frac{d\theta}{dm} = r \left( \frac{\partial \theta}{\partial s} \frac{ds}{dm} + \frac{\partial \theta}{\partial t} \frac{dt}{dm} \right) \]

\[ \frac{ds}{dm} = \frac{W_s}{W_m} \]

\[ \frac{dt}{dm} = \frac{W_t}{W_m} \]
we have

\[ W_\theta = r \left( W_s \frac{\partial \theta}{\partial s} + W_t \frac{\partial \theta}{\partial t} \right) \]

within the blade. Outside the blade,

\[
W_\theta = \begin{cases} 
\frac{\lambda}{r} - \omega r & \text{upstream of blade} \\
\frac{(r V_{\theta})_0}{r} - \omega r & \text{downstream of blade}
\end{cases}
\]

Then \( V_\theta \) and \( W \) are calculated by

\[
V_\theta = W_\theta + \omega r
\]

\[
W = \sqrt{W_\theta^2 + W_s^2 + W_t^2}
\]

The ideal density \( \rho \) is calculated by

\[
\rho = \rho'_i \left( \frac{T}{T'_i} \right)^{1/(\gamma - 1)}
\]

where \( T/T'_i \) is calculated by equation (B13) of part I (ref. 6). Then \( \partial p/\partial r \) and \( \partial I/\partial r \) are calculated by

\[
\frac{\partial p}{\partial r} = \frac{\partial p}{\partial s} \sin \varphi + \frac{\partial p}{\partial t} \cos \varphi
\]

\[
\frac{\partial I}{\partial r} = \frac{\partial I}{\partial s} \sin \varphi + \frac{\partial I}{\partial t} \cos \varphi
\]

Relative total temperature \( T'' \) is also needed and is calculated from previously calculated values of \( T''/T'_i \) and \( T'_i \). This gives all the quantities needed to calculate \( \xi \) and \( \zeta \) from equations (A2) and (A3) of part I.
After all calculations are done, the iteration number and the maximum relative change in velocity are printed. Also, if the solution is converged on velocity, the print control variables are set to 1 whenever a positive value is specified as input. This results in output being printed for each item asked for after convergence.

There are also two error messages for NEWRHO in case the velocity at some point becomes too large or if the upstream whirl is too large. Suggestions for correcting input are given in the section Error Messages in part I.

Subroutine OUTPUT

The OUTPUT subroutine calculates and prints all the major output data from MERIDL. A flow chart for OUTPUT is shown in figure 13. Depending upon the wishes of the user, OUTPUT has the potential for printing output on three separate sets of points. These points are illustrated in figure 14. Output may be obtained (1) at the orthogonal mesh points, (2) along streamlines where they are crossed by vertical orthogonal mesh lines, and (3) along streamlines where they are crossed by user-designated hub-shroud station lines. A detailed description of the output in each case is given in part I under Printed Output.

The printing of output is controlled by the iteration counter ITER and the input variables IMESH, ISLINE, and ISTATL. Because of the large volumes of output possible, it is only given at the locations requested by these variables and when ITER is an integer multiple of these variables.

No matter what the values of IMESH, ISLINE, and ISTATL, data are calculated at the orthogonal mesh points for every iteration. (Whether or not it is printed depends upon IMESH.) Output along streamlines and on station lines is then interpolated from the calculated data at the orthogonal mesh points if the values of ISLINE or ISTATL indicate that the user desires these outputs at the current iteration. Output along streamlines is also calculated if it is needed for plotting (controlled by IPLNOT) or if it is needed for calculating the input to the TSONIC program (controlled by ITSON).

The first sections of the OUTPUT routine calculate data on the orthogonal mesh. At the main entry to this routine, $W_s$, $W_t$, and $W_\theta$ are known from NEWRHO; and the other velocity components and flow angles are calculated as follows:

\[ W_m = \sqrt{W_s^2 + W_t^2} \]

\[ \sin(\alpha - \phi) = \frac{W_t}{W_m} \]
Calculate velocities and angles at mesh points, $W_m$, $\sin(\alpha-\phi)$, $\cos(\alpha-\phi)$, $W_e$, $W_r$, $\alpha$, and $\beta$

Exchange surface velocities if necessary, due to machine orientation

Is orthogonal mesh output to be printed?

If streamline output desired by user, needed for plotting, or for TSONIN

Calculate $z_e$, $r_e$, and $m_e$-coordinates of streamlines

Interpolate on orthogonal mesh to obtain $W_m$, $W_e$, $W_r$, $W_{m_e}$, $W_{e_r}$, $W_{r_m}$, $\alpha$, $\beta$, and $w$ on streamlines

Determine which streamline points lie within the blade; ILS and IIS
LININT
Interpolate on orthogonal mesh to obtain $W_r$ and $W_{tr}$ on streamlines $tr$

Is streamline output to be printed?

Yes
Write output along streamlines

No

Is hub-shroud station-line output desired by user?

Yes
Calculate $z$, $r$, and $m$-coordinates of hub-shroud station lines

LININT
Interpolate on orthogonal mesh to obtain $W_r$, $W_{tr}$, $W_{t}$, $W_{cr}$, $U/r_C$, $W_m$, $\alpha$, $\beta$, $W_t$, and $W_{tr}$ along station lines

Write output along station lines

No
TSONIC
Input desired?

Yes
TSONIN
Return

No
Return

Figure 13. Flow chart for OUTPUT.
\[
\cos(\alpha - \varphi) = \frac{W_s}{W_m}
\]

\[
W_z = W_s \cos \varphi - W_t \sin \varphi
\]

\[
W_r = W_t \cos \varphi + W_s \sin \varphi
\]

\[
\alpha = \tan^{-1}\left(\frac{W_r}{W_z}\right)
\]

\[
\beta = \tan^{-1}\left(\frac{W_\theta}{W_m}\right)
\]

This coding is followed by an entry point TOUTPT which is used only after TVELCY has been called to obtain transonic velocities (see the block diagram, fig. 2, when REDFAC < 1.0). From this entry point, the velocity components are calculated some-
what differently since $W$ has been recalculated by TVELCY, as well as $\beta$ upstream and downstream of the blade. The angle $\alpha$ is assumed to be the same as in the final subsonic iteration. With $W$, $\beta$, and $\alpha$ known, the velocity components are now calculated as follows:

\[
\begin{align*}
W_m &= W \cos \beta \\
W_\theta &= W \sin \beta \\
W_z &= W_m \cos \alpha \\
W_r &= W_m \sin \alpha \\
V_\theta &= W_\theta + \omega r
\end{align*}
\]

At this point in the program, all velocity components and flow angles have been calculated, regardless of the entry point. With velocity components and flow angles known, streamline curvature is obtained from

\[
\frac{1}{r_c} = \frac{d \alpha}{dm} = \frac{\partial \alpha}{\partial s} \cos(\alpha - \varphi) + \frac{\partial \alpha}{\partial t} \sin(\alpha - \varphi)
\]

Then critical velocity ratio is obtained from

\[
T'' = T_1' - \frac{2\omega \lambda - (\omega r)^2}{2c_p}
\]

\[
\frac{W}{W_{cr}} = \frac{W}{\sqrt{\frac{2\gamma R}{\gamma + 1} T''}}
\]

The subroutine BLDVEL is then called to calculate and return an estimate of blade surface velocities. Finally, a check is made to see if the suction- and pressure-surface velocities have to be exchanged because of the orientation of the turbine or compressor. At this point, all desired information has been calculated on the orthogonal mesh and is printed if ITER is a multiple of IMESH.
The next section of the OUTPUT routine calculates output on the streamlines where they are intersected by vertical orthogonal mesh lines. This output is calculated only if ITER is a multiple of ISLINE, IPILOT, or ITSON. First, streamline z- and r-coordinates are calculated. The m-coordinates are then calculated from these, using the z = 0 point along a streamline to correspond to m = 0. Interpolations are then made by using LININT and the orthogonal mesh data to obtain $W_z$, $W_r$, $W_\theta$, $W/W_{cr}$, and $1/r_c$. By using variations of the preceding formulas, $W_m$, $\alpha$, $\beta$, and $W$ are calculated from these values. Subroutine ILETE is called to establish which mesh points along streamlines are between the blade leading and trailing edges. Subroutine LININT is then used to obtain $W_L$ and $W_{tr}$ at these points. Finally, this output is printed if ITER is a multiple of ISLINE.

The next section of the OUTPUT routine calculates output on user-designated hub-shroud station lines where they intersect the streamlines. This output is calculated and printed in the hub-shroud direction, in contrast to the throughflow direction of the previous two sets of output. It is only calculated if ITER is a multiple of ISTATL. The z- and r-coordinates of the station lines are calculated first. All "regular" station lines are straight lines (not necessarily radial) from the hub to the shroud. "Blade edge" station lines are those whose hub and tip coordinates correspond to the intersections of the blade leading and trailing edges with the hub and tip. Coordinates along these station lines will follow these edges even when the edges are curved. After the z- and r-coordinates are established, m-coordinates are calculated from these, again using z = 0 as the reference for m = 0. Interpolations are then made using LININT with the orthogonal mesh values to obtain $W_z$, $W_r$, $W_\theta$, $W/W_{cr}$, and streamline curvature. The values of $W_m$, $\alpha$, $\beta$, and $W$ are calculated from these. LININT is then called to obtain $W_L$ and $W_{tr}$ for station lines which lie within the blade. The station-line output is then printed.

The final small section of OUTPUT calls the TSONIN subroutine to obtain input data for the TSONIC program (ref. 5). This call is only made if ITER is a multiple of ITSON.

Subroutine BLDVEL

This subroutine calculates blade-surface velocities and densities and $F_r$. First, $\partial(rV_\theta)/\partial t$ and $\partial(rV_\theta)/\partial s$ are calculated by using the SLOPES subroutine. Then, $[d(rV_\theta)/dm]B\cos \beta$ is calculated, and $W_L$ and $W_{tr}$ are calculated by equation (G4) of part I (ref. 6). From this, $\rho_L$ and $\rho_{tr}$ are calculated by equations (B13) and (D4) of part I. The average density $\rho_{av}$ is calculated by Simpson's rule:

$$\rho_{av} = \frac{\rho_L + 4\rho_{mid} + \rho_{tr}}{6}$$
This quantity is used in NEWRHO in the next iteration. Then, the predicted value of $F_r$ is calculated by

$$F_r = \frac{W}{B} \left( \frac{\partial \theta}{\partial s} \sin \varphi + \frac{\partial \theta}{\partial t} \cos \varphi \right) DFDM$$  \hspace{1cm} (4)

where

$$DFDM = -B \cos \beta \frac{\partial (rV_\theta)}{\partial m}$$

Equation (4) is obtained from equations (B23) and (G2) of part I (ref. 6). The new value for $F_r$ is calculated from the old $F_r$ and the predicted value of $F_r$ by using the input damping factor FNEW, as explained in the section INPUT of part I.

At the end, the minimum and maximum predicted values of $F_r$ and the maximum change in $F_r$ are calculated and printed. If debug output is requested, the arrays which change each iteration are printed.

Subroutine ILETE

The points where streamlines are intersected by the vertical orthogonal mesh lines are the streamline mesh points. These are, in general, different from the orthogonal mesh points. Subroutine ILETE calculates two integer arrays, ILS and ITS. They con-
tain the numbers of the vertical mesh lines at the first intersection of a streamline with a vertical mesh line inside the blade region at the leading and trailing edges of the blades. These points are illustrated in figure 15. The ILS and ITS arrays are used in OUTPUT in the calculation of blade surface velocities along streamlines.

Subroutine TSONIN

Subroutine TSONIN generates and prints the data required as input to the TSONIC blade-to-blade analysis program (ref. 5). Subroutine TSONIN is only called when ITER is a multiple of ITSON. The data generated are printed for each of the stream surfaces from hub to shroud, using 1 percent of the mass flow about a streamline to define a stream surface or flow channel.

A complete description of the TSONIC input is given in the TSONIC report (ref. 5). The output generated in TSONIN is slightly different from what is required by TSONIC. These differences and the changes which have to be made to make these data acceptable to TSONIC are described in part I.

Subroutine INDEV

Subroutine INDEV recalculates $\partial \theta / \partial s$ to allow for incidence and deviation. This means that the mid-channel flow surface differs from the blade mean camber line near the leading and trailing edges, so as to match the upstream and downstream flow angles. Figure 16 shows the procedure as applied to the leading edge. A similar correction is made at the trailing edge. A correction for blockage is made so as to satisfy both continuity and tangential momentum at blade leading and trailing edges.

The calculation starts at the hub and proceeds to successive horizontal mesh lines up to the tip. Both incidence and deviation corrections are calculated for each horizontal mesh line. The corrected $(\partial \theta / \partial s)_{bf}$ of the blade leading or trailing edge is calculated from equations (F1) and (F2) of part I (ref. 6). These equations relate $(\partial \theta / \partial s)_{bf}$ to the flow angles $\beta_{bf}$ and $\beta_{fs}$.

The corrections to $\partial \theta / \partial s$ are made so that the difference varies linearly from the blade leading or trailing edge for the distance specified in appendix F of part I. After the corrections are made, the incidence and deviation angles are printed if requested.

No correction is made to $\partial \theta / \partial t$ since it is nearly normal to the flow.
Subroutine SLPLOT

Subroutine SLPLOT makes a microfilm plot of the streamlines in the hub-shroud meridional flow plane. The first small section of coding plots a separate frame of film identifying either a subsonic solution and iteration number or a transonic solution. The remaining coding plots the hub, shroud, and blade profiles in the meridional plane and then adds the streamlines to the same plot. An example of this plot is given in figure 16(j) of part 1.

Subroutine SVPLOT

Subroutine SVPLOT makes microfilm plots of relative velocities on all streamlines from hub to shroud. These plots are only made when ITER is a multiple of IPLOT or when ITER = 1. A separate plot is made for the velocities on each streamline. These plots include mean flow velocities and blade surface velocities plotted against meridional coordinates. Examples of these plots are given in figures 16(k) and (l) of part I (ref. 6). After the separate streamline plots are made, three composite plots are made. The first contains the mean flow velocities for all streamlines. The second and third contain the suction- and pressure-surface velocities, respectively, for all streamlines.

Subroutine TVELCY

Subroutine TVELCY calculates the full-mass-flow, transonic solution when REDFAC is less than 1. The velocity-gradient equation developed in appendix C of part I is used to obtain the solution. Figure 17 is a flow chart for TVELCY.
Start

REDFAC = 1

Yes

No

Yes

Set JZ = 2 to get supersonic solution

No

Restore full-mass-flow values and reinitialize subroutines LAMDAF and RVHTA

Calculate \( \frac{dW_{mm}}{dm} \) and \( \frac{dW_g}{dm} \)

I = 1

60

First estimate for WHUB

Calculate a, b, and d for velocity-gradient equation (A7) of part 1; set c = 0

Calculate c for velocity-gradient equation; calculate arrays for

\[ 2\alpha - \text{ln} \alpha^2, \]
\[ 2\alpha - \text{ln} \alpha^2, \]
\[ p \cos \alpha - \rho \alpha B \]

Calculate e and f for velocity-gradient equation

Solve velocity-gradient equation; calculate mass flow

Integrated mass flow converged?

Yes

No

Iteration
Call CONTIN to get WHUB

Proceeding OK?

No

(Over 1000 iterations)

Choked?

Yes

Calculate minimum choking mass flow

No

Over 1000 iterations?

Yes

Choked?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?

Yes

Have free-stream values changed?

No

Over 1000 iterations?
The first step in the program is to restore the full value of mass flow, rotational speed, and inlet and outlet whirl. The subroutines LAMDAF and RVTHTA must then be reinitialized.

Next, \( \partial W_m / \partial m \) and \( \partial W_\theta / \partial m \) are calculated. These are calculated from the partials with respect to \( s \) and \( t \) by using the angle \( \alpha - \varphi \). Since the calculations are based on the reduced-mass-flow values of \( W_m \) and \( W_\theta \), the result must be divided by REDFAC to obtain the full-mass-flow values.

Statement 60 is the beginning of the outer DO loop. It starts at the upstream boundary and solves the velocity-gradient equation for each vertical mesh line. The initial estimate of \( W \) on the hub (WHUB) is set equal to the reduced-mass-flow value for \( W \) divided by REDFAC. For a given vertical line, the coefficients \( a, b, c, d, e, \) and \( f \) of the velocity-gradient equation (A7) of part I are calculated. The coefficients are calculated by equations (A8) to (A11) of part I. Of these coefficients, \( a, b, \) and \( d \) will not be changed after the initial calculation, so they are calculated first. The initial arrays for whirl, temperature, and density are calculated at the same time.

For each vertical mesh line, an inner and an outer iteration is required. Each outer iteration consists of solving the velocity-gradient equation for a given distribution of upstream and downstream flow conditions. The inner iteration solves the velocity-gradient equation by varying \( W_{\text{hub}} \) at each inner iteration until continuity is satisfied. The outer iteration starts at statement 90. None of the coefficients change during the inner iteration, so the remaining coefficients, \( c, e, \) and \( f \), are calculated from equations (A9) to (A11) of part I before starting the inner iteration. Also, part of the integrand for the mass flow integration is calculated now. This part is RCARB, which is equal to \( \rho_1 \cos(\alpha - \varphi)rB \).

At statement 140, the inner iteration starts. First, initial values are set. The numerical solution of the velocity-gradient equation and the mass flow integration are done in the DO 200 loop. Trial values of WHUB are used in the velocity-gradient equation, until the solution obtained results in the input mass flow across the vertical mesh line. The first iteration will use the value calculated by the statement after statement 60. Later iterations will use estimated values calculated by CONTIN. Once WHUB is specified, the numerical solution to the velocity-gradient equation is calculated by the Heun method, as described for VBDRY. The solution procedure is the same, except that \( dW \) in equation (1) is evaluated by equation (A7) of part I (ref. 6). The mass flow is calculated by trapezoidal integration of

\[
W = \int_0^{\text{tip}} \rho W \cos \beta \cos(\alpha - \varphi)rB \, dt \tag{5}
\]
As explained in appendix D of part I (ref. 6), $\rho$ is the ideal density and $B$ is reduced to reflect any loss of stagnation pressure.

The inner iteration ends when the velocity-gradient solution gives the correct mass flow in equation (5). (If the correct mass flow is not obtained in 100 iterations, an error message is printed, and the program goes to the next vertical line.) After the end of the inner iteration, at statement 250, the upstream and downstream flow conditions are checked. If there is a significant change in the value of inlet or outlet stagnation temperature or density, or whirl, these values will be adjusted and the inner iteration will be repeated by going back to statement 90, unless there has been a total of over 1000 iterations for a given vertical mesh line. The outer iteration is completed when there is no significant change in the solution, and the program goes to the next vertical line (the DO 280 loop). After all vertical lines have been completed, control is returned to the main program. If the blade is choked, a message is printed with the choking mass flow.

Function TOPF

Function TOPF calculates downstream stagnation temperature $T'_D$ from the upstream stagnation temperature and the change in whirl. That is,

$$T'_0 = T'_D + \frac{\omega \left( (rV\theta)_o - \lambda \right)}{c_p}$$

The input argument (SF) is the value of the stream function (between 0 and 1). The function TOPF is then $T'_0$ for this streamline.

Functions TIPF, RHOIPF, LAMDAF, RHOOPF, and RVTHTA

These five routines are similar. Their purpose is to calculate one of the free-stream quantities as a function of stream function. Interpolation is by means of a spline fit curve.

All these subroutines have an alternate entry point for initialization. The initializing call results in a SPLINE call to calculate the coefficients for the spline fit.

If the free-stream quantities are not given as input as a function of stream function (i.e., if LSFR = 1), the stream function is first estimated and later iterated to be adjusted to the correct stream-function value. These adjustments to the stream function (SFIN and SFOUT) are done in LAMDAF and RVTHTA.
The input argument for all these subroutines is \( SF \), which is the value of the stream function.

**Subroutine CONTIN**

Subroutine CONTIN is a curve-fitting routine. On each call the calling programs must furnish a point on the curve, and then CONTIN will specify the next value of the abscissa. The calling program must then calculate the ordinate corresponding to this abscissa. After three calls, a parabola is fitted through the three points, and this is used to estimate the abscissa where the desired ordinate will be obtained. \( XEST \) is the value of the abscissa, and \( YCALC \) is the value of the ordinate on each call. \( XEST \) is changed by CONTIN to return the next value of the abscissa to the calling program. Figure 18 is a flow chart for CONTIN. Flow through the program is controlled by the value of \( IND \). For each new case, \( IND \) is set to 1 by the calling program. Then CONTIN changes the value of \( IND \) on later calls. The significance of \( IND \) on the various calls is given in table II. \( XDEL \) is the maximum increment for the change in \( XEST \). On the first two calls, usually \( XEST \) is increased by \( XDEL \) each time. The exception is when \( YCALC \) is greater than \( YGIV \) and the subsonic solution is desired (\( JZ = 1 \)). Then \( XEST \) is decreased by \( XDEL \) each time.

On the third and later calls, there are always three points so that a parabola can be fitted through the three points. The parabolic coefficients are calculated by subroutine PABC. Anytime that \( XEST \) falls outside the range of previously calculated values, a shift is made until \( XEST \) is within the desired range.

When the parabolic curve is close to a straight line, equation (D13) is used instead of the quadratic formula. The reason for this is explained in appendix D.

Figure 19 illustrates the procedure for a typical case. On the first call to CONTIN, \( IND = 1 \) and \( YCALC \) corresponding to \( XEST \) is furnished by the calling program. Suppose that \( YCALC \) is less than \( YGIV \) and that the subsonic solution is requested. Then \( XEST \) becomes \( XORIG \), and \( YCALC \) becomes \( Y(1) \) in figure 19. \( XORIG \) will be the origin for the curve fitting so that \( X(1) = 0 \) in this case. Next CONTIN increases \( XEST \) by \( XDEL \). Then a return is made to the calling program to obtain the \( YCALC \) which corresponds to this value of \( XEST \). On the second call to CONTIN, the new value of \( YCALC \) becomes \( Y(2) \) and \( XEST - XORIG \) becomes \( X(2) \), as indicated in figure 19. Subroutine CONTIN increases \( XEST \) by \( XDEL \) again, and a return is made to obtain \( YCALC \) for the third time. On the third call to CONTIN, the new value of \( YCALC \) becomes \( Y(3) \) and \( XEST - XORIG \) becomes \( X(3) \). This gives the three points shown in figure 19. The curve shown represents the true curve of \( YCALC \) against \( XEST \).

At this time, a check is made to determine whether the solution is within the range
Figure 18. - Flow chart for CONTIN.
TABLE II. - SIGNIFICANCE OF IND IN VARIOUS CALLS TO CONTIN

<table>
<thead>
<tr>
<th>Value of IND</th>
<th>Call</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First</td>
<td>First call</td>
</tr>
<tr>
<td>2</td>
<td>Second</td>
<td>JZ = 1, YCALC less than WTFL, or JZ = 2</td>
</tr>
<tr>
<td>3</td>
<td>Second</td>
<td>JZ = 1 and YCALC greater than WTFL</td>
</tr>
<tr>
<td>4</td>
<td>Third</td>
<td>IND = 2 on second call</td>
</tr>
<tr>
<td></td>
<td>Fourth or later</td>
<td>Right shift was made so that XEST will be within range of stored previous values.</td>
</tr>
<tr>
<td>5</td>
<td>Third</td>
<td>IND = 3 on second call</td>
</tr>
<tr>
<td></td>
<td>Fourth or later</td>
<td>Left shift was made so that XEST will be within range of stored previous values.</td>
</tr>
<tr>
<td>6</td>
<td>Fourth or later</td>
<td>Subsonic or supersonic solution is predicted by quadratic fit and is within range of solutions obtained.</td>
</tr>
<tr>
<td>7</td>
<td>Fourth or later</td>
<td>Choked flow is predicted by quadratic fit and is within range of solutions obtained.</td>
</tr>
<tr>
<td>10</td>
<td>Never</td>
<td>Choked solution found</td>
</tr>
<tr>
<td>11</td>
<td>Never</td>
<td>100 calls made but no solution found</td>
</tr>
</tbody>
</table>

Figure 19. - Starting procedure for CONTIN.
of the three points obtained. If not, additional points are calculated, and the three points are shifted as required. For example, in figure 19, a shift to the right is required. In this case, point 2 would become point 1, point 3 would become point 2, and XEST would be increased by XDEL. This procedure is repeated until either the solution or the maximum point is within the range of the three points obtained.

Since the curve represents mass flow as a function of the velocity at some point, the curve will be of the type shown. The maximum point on the curve is the choking mass flow. This type of curve is approximated well by a quadratic curve. After it has been determined that a solution is within the range of the three points (i.e., Y(1) ≤ YGIV ≤ Y(3) for a subsonic solution), a parabola is fitted through the three points. This situation is illustrated in figure 20. The next value of XEST is determined by the point where the parabolic curve intersects the YGIV line. Then the return is made to obtain YCALC. If YCALC is sufficiently close to YGIV, this will be the solution. Otherwise, CONTIN is called again, XEST - XORIG becomes X(2), YCALC becomes Y(2), and the procedure is repeated (as many as 100 times) until YCALC is sufficiently close to YGIV.

The detailed operation of subroutine CONTIN is given in figure 18 and table II. The calling statement for CONTIN is

```
CALL CONTIN(XEST, YCALC, IND, JZ, YGIV, XDEL)
```

The input variables for CONTIN are

- **XEST** last value of X used to calculate YCALC
- **YCALC** value of Y corresponding to XEST (calling program calculates YCALC)
IND controls sequence of calculation in CONTIN; calling program sets IND = 1 to indicate a new solution

JZ determines whether subsonic or supersonic solution will be obtained:
JZ = 1, subsonic solution
JZ = 2, supersonic solution

YGIV value of Y desired for solution
XDEL maximum permissible change in XEST between iterations

The output variables for CONTIN are
XEST value of X to be used to calculate the next value for YCALC
IND used to control next iteration in CONTIN and to indicate when a choked solution is found or when no solution can be found (table II)

The internal variables for CONTIN are
ACB2 \( \frac{a(c - y)}{b^2} \)
APA coefficient \( a \) of \( X^2 \) in quadratic fit
BPB coefficient \( b \) of \( X \) in quadratic fit
CPC constant \( C \) in quadratic fit
DISCR discriminant, \( \sqrt{b^2 - 4ac} \)
NCALL number of times CONTIN has been called for a given case
X array of three values of XEST - XORIG
XORIG value of XEST on initial call, modified by right or left shifts
XOSHFT amount of change of XORIG
Y array of three values of YCALC

Subroutine PABC

Subroutine PABC calculates coefficients A, B, and C of the parabola \( y = Ax^2 + Bx + C \) passing through three given \( X, Y \) points.

Subroutine INRSCT

Subroutine INRSCT calculates the coordinates of the point of intersection of two
spline curves lying on a common plane which are known to cross within the range of the end points of each. In a general x-y coordinate system, the first spline curve is supplied to INRSCT as a function of \( x \)

\[ y = f(x) \]

and the second as a function of \( y \)

\[ x = g(y) \]

The solution technique consists of systematically constructing pairs of tangent slopes to the two curves and locating the points of intersection of the two slopes. Each intersection point provides new coordinates from which new slopes and an intersection are calculated. These intersections quickly converge to the intersection point of the original curves.

This technique is illustrated in figure 21. The original trial x-coordinate is always midway between the end points for \( f(x) \). This value is \( x_1 \), from which \( y_1 \) and slope \( s_1 \) are calculated by SPLINT. The calculated \( y_1 \) is then used as input to SPLINT for \( g(y) \). From this SPLINT call, \( x_2 \) and \( s_2 \) are calculated, as shown in figure 21. The intersection point of the two slopes is calculated from
\[ x_c = x_2 + \frac{s_1 s_2 (x_2 - x_1)}{1 - s_1 s_2} \]

\[ y_c = y_1 + \frac{s_1 (x_2 - x_1)}{1 - s_1 s_2} \]

Then \( x_c \) becomes \( x_1 \) for the following iteration of this process.

To check convergence of this process, the distance is calculated between each pair of intersection points \( x_c, y_c \) for adjacent iterations. When this distance becomes less than the tolerance, an exit is made from INRSCT. Failing to meet the tolerance in 20 iterations causes an error message to be printed.

The calling statement for subroutine INRSCT is

```
CALL INRSCT(XCURV1, YCURV1, N1, XCURV2, YCURV2, N2, XCROSS, YCROSS)
```

The input arguments for INRSCT are

- \( XCURV1(N1) \) x-coordinates for \( f(x) \)
- \( YCURV1(N1) \) y-coordinates for \( y = f(x) \)
- \( XCURV2(N2) \) x-coordinates for \( x = g(y) \)
- \( YCURV2(N2) \) y-coordinates for \( g(y) \)
- \( N1 \) number of spline points for \( f(x) \)
- \( N2 \) number of spline points for \( g(y) \)

The output arguments for INRSCT are

- \( XCROSS \) x-coordinate of intersection of two input curves
- \( YCROSS \) y-coordinate of intersection of two input curves

Subroutine ROOT

Subroutine ROOT finds a root for \( f(x) = y \) by the bisection method. The function \( f(x) \) must be defined on the interval \([a, b]\) by the subroutine FUNCT. FUNCT is a dummy name; any subroutine name may be used in the calling program. In MERIDL, FUNCT is CROSCD.

The interval is bisected 20 times by ROOT. This gives a resolution of \( x \) of \( 10^{-6} \) times the interval length. After the root has been located, the difference \( f(x) - y \) is
checked to see if it is less than TOLERY. If not, a message is printed with details on the iterated calculations.

The calling statement for ROOT is

\[ \text{CALL ROOT}(A, B, Y, \text{FUNCT}, \text{TOLERY}, X, \text{DFX}) \]

The input arguments for ROOT are

- \( A \)  
- \( B \)  
- \( Y \)  
- \( \text{FUNCT} \) external subroutine to calculate \( f(x) \)
- \( \text{TOLERY} \) tolerance on solution (\( x \) is accepted as a root if \( |f(x) - y| < \text{TOLERY} \))

The output arguments for ROOT are

- \( X \) value at \( x \) such that \( f(x) = y \)
- \( \text{DFX} \)  

The calling sequence for FUNCT must be

\[ \text{FUNCT}(X, FX, DFX,) \]

These arguments are defined as follows:

- \( X \)  
- \( \text{FX} \)  
- \( DFX \)  

Subroutine LININT

Subroutine LININT is a general-purpose subroutine for two-dimensional interpolation. It is called many times by several subroutines.

Subroutine LININT locates the point \((x_o, y_o)\) in a two-dimensional mesh with coordinates stored in the \( x \) and \( y \) arrays. Then the value of \( z_o \) at \( x_o, y_o \) is interpolated from the \( z \)-array values corresponding to the \( x \) and \( y \) arrays. Figure 22 is a flow chart for LININT.
Start

10 Change I or J if necessary

Calculate direction to move to quadrilateral

Quadrilateral found?

Yes

IJEX = 1

No

Adjust I, J

Check for extrapolation

Compute constants for eq. (D8)

30 Compute a, b, and c for eq. (D7) or (D9)

Yes

Error message

Return

No

110 $b^2 - 4ac < 0$?

Yes

Linear?

Yes

Compute FF by eq. (D13)

No

No

Compute both quadratic roots

FF is set to correct root

80

Yes

IJEX = 2?

Yes

IJEX = 2

FY = FF

No

IJEEX = 2

FX = FF

Compute $Z_0$

100 Compute constants for eq. (D10) by interchanging corners

90

No

100

Return

Figure 22. - Flow chart for LININT.
A typical mesh is shown in figure 23. The mesh need not be orthogonal; but it must consist of two sets of lines, with one set running more or less horizontally (never vertical) and the other set running more or less vertically (never horizontal). The number of vertical lines is NX, and I denotes the number of the line (running from 1 at the left to NX at the right). The number of horizontal lines is NY, and J denotes the number of the line (running from 1 at the bottom to NY at the top). The lines between mesh points are assumed to be straight lines.

At the outset, some value of I and J must be specified. Any value within the prescribed limits is legal. On repeated calls to LININT, usually the value from the preceding call is used. The values of I and J desired are the numbers shown at the bottom of figure 23. In this figure I = 4, J = 3. The procedure is to check to see on which side of each of the four boundary lines the point lies. The variables ABOVE and RIGHT are used to indicate the position. ABOVE = -1 indicates the point is below the bottom line, ABOVE = 0 the point is between the bottom and top lines, and ABOVE = 1 the point is above the top line. Similarly, RIGHT = -1 indicates the point is to the left of the left line, RIGHT = 0, the point is between the left and right lines, and RIGHT = 1 the point is to the right of the right line. Thus, when ABOVE = RIGHT = 0, we have the correct mesh region. If not, I and/or J are incremented by plus or minus 1 to move to the proper adjacent region. In this way, eventually the proper region will be found. If the point lies entirely outside the region defined, the nearest mesh region to the point \((x_0, y_0)\) will be found. In this case, extrapolation is required, and the variable EXTRAP is used to indicate the direction of extrapolation. EXTRAP is dimensioned 2. EXTRAP(1) corresponds to ABOVE, and EXTRAP(2) to RIGHT.

After the proper mesh-point region is found, interpolation between the function val-
ues at the four corners is used. The method used is described in appendix D. First, the quadratic coefficients are calculated by equation (D8) or (D10). Then, the quadratic equation (D7) or (D9) is solved either by the quadratic formula, or by the binomial expansion, equation (D13), as explained in appendix D.

The same coding is used to calculate both $f_x$ and $f_y$. After these values are obtained, equation (D14) is used to calculate the interpolated value of $z_o$.

The calling statement for LININT is

```plaintext
CALL LININT(X, Y, Z, NX, NY, NDIMX, NDIMY, XO, YO, Z0, I, J)
```

The input variables for LININT are

- **X**: two-dimensional array of x-coordinates of mesh points
- **Y**: two-dimensional array of y-coordinates of mesh points
- **Z**: two-dimensional array of z-function values at mesh points
- **NX**: number of mesh points in the x-direction
- **NY**: number of mesh points in the y-direction
- **NDIMX**: dimension of X, Y, and Z arrays in the x-direction
- **NDIMY**: dimension of X, Y, and Z arrays in the y-direction
- **XO**: x-coordinate of interpolation point
- **YO**: y-coordinate of interpolation point
- **I**: initial guess at number of vertical mesh line to the left of (XO, YO)
- **J**: initial guess at number of horizontal mesh line below (XO, YO)

The output variables for LININT are

- **Z0**: interpolated value of Z at (XO, YO)
- **I**: number of vertical mesh line to the left of (XO, YO)
- **J**: number of horizontal mesh line below (XO, YO)

The internal variables for LININT are

- **ABOVE**: integer, 1 indicates (XO, YO) is above the current I, J region, 0 within, and -1 below
- **ACB2**: $ac/b^2$ (eq. (D13))
- **CASE**: used to indicate whether F1 or F2 is the proper solution
- **DISCR**: discriminate, $b^2 - 4ac$ (eq. (D7) or (D9))
EXTRAP array to indicate extrapolation either horizontally or vertically
FA \(-b/2a\) (eq. (D7) or (D9))
FB \(\sqrt{(b^2 - 4ac)/2a}\) (eq. (D7) or (D9))
FF \(f_x\) or \(f_y\)
FX \(f_x\)
FY \(f_y\)
F1 \(\left(-b - \sqrt{b^2 - 4ac}\right)/2a\)
F2 \(\left(-b + \sqrt{b^2 - 4ac}\right)/2a\)
IJEX indicator, first or second pass through coding to calculate \(f_x\) or \(f_y\)
IN new value for I
JN new value for J
QA \(a\) (eq. (D8) or (D10))
QB \(b\) (eq. (D8) or (D10))
QC \(c\) (eq. (D8) or (D10))
RIGHT integer, 1 indicates \(X0, Y0\) is to the right of the current I, J region, 0 within, and -1 left
X01 \(x_{01}\) (see appendix D for notation)
X02 \(x_{02}\) or \(x_{03}\)
X13 \(x_{13}\) or \(x_{12}\)
X21 \(x_{21}\) or \(x_{31}\)
X42 \(x_{42}\) or \(x_{43}\)
Y01 \(y_{01}\)
Y02 \(y_{02}\) or \(y_{03}\)
Y13 \(y_{13}\) or \(y_{12}\)
Y21 \(y_{21}\) or \(y_{31}\)
Y42 \(y_{42}\) or \(y_{43}\)
Subroutine SPLINE

Subroutine SPLINE calculates the first and second derivatives of a cubic spline curve at the spline points. SPLINE solves a tridiagonal matrix given in reference 12 to obtain the coefficients for the piecewise cubic polynomial function giving the spline fit curve. The SPLINE routine is based on the end-point condition that the second derivative at either end point is one-half that of the next spline point.

The calling statement for SPLINE is

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

The input variables for SPLINE are
X array of ordinates
Y array of function values corresponding to X
N number of X and Y values given

The output variables for SPLINE are
SLOPE array of first derivatives
EM array of second derivatives

Subroutine SPLINT

Subroutine SPLINT is used for interpolation, including interpolation of the derivative. The interpolation is based on the cubic spline curve, with the same end conditions as SPLINE. The alternate entry point, SPLENT, allows for interpolation at a new set of points based on the spline curve of the previous SPLINE call.

The input variables for SPLINT are
X array of spline point ordinates
Y array of function values at spline points
N number of X and Y values given
Z array of ordinates at which interpolated values and derivatives are desired
MAX number of Z values given

The output variables for SPLINT are
YINT array of interpolated function values
DYDX array of interpolated derivatives
Subroutine SLOPES

Subroutine SLOPES calculates the first derivatives (slopes) based on a parabolic fit through three adjacent points. This subroutine is used when the input points may not be sufficiently smooth for the SPLINE subroutine.

The calling statement for subroutine SLOPES is

```
CALL SLOPES(X, Y, N, SLOPE)
```

The input arguments for SLOPES are

- X: array of ordinates
- Y: array of function values corresponding to X
- N: number of X and Y values given

The output variable for SLOPES is

- SLOPE: array of first derivatives

MAIN DICTIONARY

The main dictionary for MERIDL is given in this section. It contains the definitions of variables for all the principal subroutines (from INPUT to RVTHTA, see table of contents) of the program. The remaining subroutines (CONTIN to SLOPES) are of a general-purpose nature and have their own local dictionaries included in their descriptions.

All important variables are included in the main dictionary. These include all COMMON variables, any dimensioned variables in the subroutines, and all important undimensioned variables. Only locally used undimensioned variables of minor importance are not included.

The names of all dimensioned variables are followed by the variables which determine what the dimensions should be. For example, the three-dimensional array A is dimensioned A(4, 100, 101) in the /VARCOM/ COMMON but is listed as A(4, MM, MHTP1) in the dictionary. This enables the user to easily alter the dimension of A (and reduce the program's variable storage) if he knows maximum limits to MM and MHTP1 for his application. See the section STORAGE REQUIREMENTS for further explanation.

The dictionary also indicates the COMMON blocks or the subroutines in which each variable is used. Variables in COMMON are used in many subroutines. The COMMON blocks are listed for each subroutine in table I.
<table>
<thead>
<tr>
<th>Variable name</th>
<th>COMMON block</th>
<th>Subroutine</th>
<th>Description and comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>MESHO</td>
<td>left-hand boundary on an interval of z-coordinate, m</td>
</tr>
<tr>
<td>A(4, MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>coefficients of finite-difference equation (A7) for stream function, u</td>
</tr>
<tr>
<td>A0</td>
<td></td>
<td>COEF</td>
<td>a_0 (eq. (A8))</td>
</tr>
<tr>
<td>AA(MHTP1)</td>
<td></td>
<td>VBDRY</td>
<td>coefficients, a, of velocity-gradient equation ((C9), part 1)</td>
</tr>
<tr>
<td>AAA(NHUB)</td>
<td></td>
<td>MESHO</td>
<td>dummy array of slopes of a spline fit of horizontal rows in RRAD array</td>
</tr>
<tr>
<td>AAA(100)</td>
<td></td>
<td>PTBDRY</td>
<td>dummy array of slopes of spline fit curves</td>
</tr>
<tr>
<td>AAA(MHTP1 or MM)</td>
<td></td>
<td>NEWRHO</td>
<td>dummy array used in SPLINE calls</td>
</tr>
<tr>
<td>AAA(MHTP1 or NOSTAT or NSL or 20)</td>
<td>OUTPUT</td>
<td>dummy array used in SPLINT calls</td>
<td></td>
</tr>
<tr>
<td>AAA(NIN or NOUT)</td>
<td></td>
<td>LAMDAF</td>
<td>dummy array used in SPLINT calls</td>
</tr>
<tr>
<td>AAA(NIN or NOUT)</td>
<td></td>
<td>RVHTA</td>
<td></td>
</tr>
<tr>
<td>AANDK(integer variable)</td>
<td></td>
<td>TSONIN</td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>ALPHA(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>alpha at orthogonal mesh points, rad</td>
</tr>
<tr>
<td>ALPHLE</td>
<td></td>
<td>INDEV</td>
<td>alpha_e, rad</td>
</tr>
<tr>
<td>ALPHTE</td>
<td></td>
<td>INDEV</td>
<td>alpha_t, rad</td>
</tr>
<tr>
<td>ALPSL(MM, NSL)</td>
<td></td>
<td>SLCOM</td>
<td>alpha at points along streamlines where they cross vertical mesh lines, rad</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>ALPST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$\alpha$ at points along station lines where they cross streamlines, rad</td>
</tr>
<tr>
<td>ALVERT(MHTP1)</td>
<td>OUTPUT</td>
<td></td>
<td>temporary storage for values of $\alpha$ from ALPHA array on vertical mesh lines, rad</td>
</tr>
<tr>
<td>ANG(NPPP)</td>
<td>INPLOT</td>
<td>THIKOM</td>
<td>TSONIN</td>
</tr>
<tr>
<td>ANGR(11, NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>angles with respect to radius of hub-shroud lines of alternate mesh (fig. 26), rad</td>
</tr>
<tr>
<td>ANGZ(11, NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>angles with respect to z-axis of input blade sections at alternate mesh points (fig. 26), rad</td>
</tr>
<tr>
<td>AR</td>
<td>INPUTT</td>
<td></td>
<td>input gas constant, $R$, J/(kg)(K)</td>
</tr>
<tr>
<td>ATVEL(MHTP1)</td>
<td>TVELCY</td>
<td></td>
<td>coefficients, $a$, of velocity-gradient equation ((A7), part I) at orthogonal mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>B</td>
<td>MESHO</td>
<td></td>
<td>right-hand boundary on interval of z-coordinate, m</td>
</tr>
<tr>
<td>BB(MHTP1)</td>
<td>VBDRY</td>
<td></td>
<td>coefficients, $b$, of velocity-gradient equation ((C9), part I)</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BDY(4)</td>
<td>VBDRY</td>
<td></td>
<td>variable containing words INLET and OUTLET used in printing error message</td>
</tr>
<tr>
<td>BESP(MM)</td>
<td>TSONIN</td>
<td></td>
<td>normal thicknesses of a stream channel, at MR,RMSP points, printed as input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>BETA(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$\beta$ at orthogonal mesh points, rad</td>
</tr>
<tr>
<td>BETAI</td>
<td>TSONIN</td>
<td></td>
<td>$\beta_1$ at blade leading edge (input to TSONIC, ref. 5), deg</td>
</tr>
<tr>
<td>BETAO</td>
<td>TSONIN</td>
<td></td>
<td>$\beta_0$ at blade trailing edge (input to TSONIC, ref. 5), deg</td>
</tr>
<tr>
<td>BETSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$\beta$ at points along streamlines where they cross vertical mesh lines, rad</td>
</tr>
<tr>
<td>BETST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$\beta$ at points along station lines where they cross streamlines, rad</td>
</tr>
<tr>
<td>BLDAT(integer variable)</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>BLDCRD</td>
<td>INDEV</td>
<td></td>
<td>true blade chord along a horizontal mesh line, m</td>
</tr>
<tr>
<td>BLDEV</td>
<td>INDEV</td>
<td></td>
<td>deviation angle, corrected for blockage, where a horizontal mesh line intersects trailing edge $(\beta_{bl} - \beta_{b})_{te}$, deg</td>
</tr>
</tbody>
</table>

62
<table>
<thead>
<tr>
<th>Variable name</th>
<th>COMMON block</th>
<th>Subroutine</th>
<th>Description and comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLINC</td>
<td>INDEV</td>
<td></td>
<td>incidence angle, corrected for blockage, where a horizontal mesh line intersects leading edge ( (\beta_{bf} - \beta_{b})_{le} ), deg</td>
</tr>
<tr>
<td>BLNK</td>
<td>INPLOT</td>
<td></td>
<td>blank word used in some plot titles</td>
</tr>
<tr>
<td>BRNG</td>
<td>INPLOT</td>
<td></td>
<td>bottom or lower range of values on a given plot</td>
</tr>
<tr>
<td></td>
<td>SVPLOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTABLD</td>
<td>INDEV</td>
<td></td>
<td>blade mean camber line angle at leading or trailing edge, ( \beta_b ), rad</td>
</tr>
<tr>
<td>BTFSLE</td>
<td>INDEV</td>
<td></td>
<td>upstream flow angle, ( \beta_f ), extrapolated linearly along a horizontal mesh line to blade leading edge, rad</td>
</tr>
<tr>
<td>BTFSTE</td>
<td>INDEV</td>
<td></td>
<td>downstream flow angle, ( \beta_f ), extrapolated linearly along a horizontal mesh line back to blade trailing edge, rad</td>
</tr>
<tr>
<td>BTH(MM, MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>B at orthogonal mesh points, rad (These values are corrected for total pressure loss through the blade row.)</td>
</tr>
<tr>
<td>BTHLE</td>
<td>INDEV</td>
<td></td>
<td>( B_{le} ), rad</td>
</tr>
<tr>
<td>BTHSL</td>
<td>TSONIN</td>
<td></td>
<td>B along a streamline, rad</td>
</tr>
<tr>
<td>BTHTE</td>
<td>INDEV</td>
<td></td>
<td>( B_{te} ), rad</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>BTVEL(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>coefficients, $b$, of velocity-gradient equation $((A7)$, part I) at orthogonal mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>C1</td>
<td>COEF</td>
<td></td>
<td>$c_1$ (eq. (A8))</td>
</tr>
<tr>
<td>C2</td>
<td>COEF</td>
<td></td>
<td>$c_2$ (eq. (A8))</td>
</tr>
<tr>
<td>CAMP(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$\cos(\alpha - \varphi)$ at orthogonal mesh points</td>
</tr>
<tr>
<td>CAMPLE</td>
<td>INDEV</td>
<td></td>
<td>$\cos(\alpha - \varphi)_{le}$</td>
</tr>
<tr>
<td>CAMPTE</td>
<td>INDEV</td>
<td></td>
<td>$\cos(\alpha - \varphi)_{te}$</td>
</tr>
<tr>
<td>CBETA</td>
<td>TVELCY</td>
<td></td>
<td>$\cos \beta$</td>
</tr>
<tr>
<td>CCA(MHTP1)</td>
<td>VBDRY</td>
<td></td>
<td>coefficients, $c_a$, of velocity-gradient equation $((C9)$, part I)</td>
</tr>
<tr>
<td>CCB(MHTP1)</td>
<td>VBDRY</td>
<td></td>
<td>coefficients, $c_p$, of velocity-gradient equation $((C9)$, part I)</td>
</tr>
<tr>
<td>CHANGE</td>
<td>SOR</td>
<td></td>
<td>change in value of stream function at a mesh point during an overrelaxation iteration</td>
</tr>
<tr>
<td>CHFL</td>
<td>TVELCY</td>
<td></td>
<td>choking mass flow for a vertical orthogonal mesh line, kg/sec</td>
</tr>
<tr>
<td>CHLIM</td>
<td>TVELCY</td>
<td></td>
<td>minimum choking mass flow per passage, kg/sec</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>CHORDF</td>
<td></td>
<td>TSONIN</td>
<td>length of blade section along streamline in m-direction (input to TSONIC, ref. 5), m</td>
</tr>
<tr>
<td>COSBTA</td>
<td></td>
<td>VBDRY</td>
<td>cos $\beta$</td>
</tr>
<tr>
<td>CP</td>
<td></td>
<td>CALCON</td>
<td>$c_p$, J/(kg)(K)</td>
</tr>
<tr>
<td>CPHI(MM, MHTP1)</td>
<td></td>
<td>CALCON</td>
<td>cos($\varphi$) at orthogonal mesh points</td>
</tr>
<tr>
<td>CPHILE</td>
<td></td>
<td>INDEV</td>
<td>cos($\varphi_{le}$)</td>
</tr>
<tr>
<td>CPHITE</td>
<td></td>
<td>INDEV</td>
<td>cos($\varphi_{le}$)</td>
</tr>
<tr>
<td>CPTIP(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>$2c_{p,T_i}$ at orthogonal mesh points along vertical mesh lines, (N)(m)/kg</td>
</tr>
<tr>
<td>CVEL(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>coefficients, c, of velocity-gradient equation ((A7), part 1) at orthogonal mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>CURV(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>$1/r_c$ at orthogonal mesh points, 1/m</td>
</tr>
<tr>
<td>CURVH</td>
<td></td>
<td>VBDRY</td>
<td>CURVHI or CURVHO, 1/m</td>
</tr>
<tr>
<td>CURVHI</td>
<td></td>
<td>CALCON</td>
<td>curvature of hub at point where it is intersected by first (upstream) vertical orthogonal mesh line, 1/m</td>
</tr>
<tr>
<td>CURVHO</td>
<td></td>
<td>CALCON</td>
<td>curvature of hub at point where it is intersected by last (downstream) vertical orthogonal mesh line 1/m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>CURVSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$1/r_c$ at points along streamlines where they cross vertical orthogonal mesh lines, $1/m$</td>
</tr>
<tr>
<td>CURVST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$1/r_c$ at points along station lines where they cross streamlines, $1/m$</td>
</tr>
<tr>
<td>CURVT</td>
<td>VBDRY</td>
<td></td>
<td>CURVTI or CURVTO, $1/m$</td>
</tr>
<tr>
<td>CURVTI</td>
<td>CALCON</td>
<td></td>
<td>curvature of shroud at point where it is intersected by first (upstream) vertical orthogonal mesh line, $1/m$</td>
</tr>
<tr>
<td>CURVTO</td>
<td>CALCON</td>
<td></td>
<td>curvature of shroud at point where it is intersected by last (downstream) vertical orthogonal mesh line, $1/m$</td>
</tr>
<tr>
<td>D1</td>
<td>COEF</td>
<td></td>
<td>$d_1$ (eq. (A8))</td>
</tr>
<tr>
<td>D2</td>
<td>COEF</td>
<td></td>
<td>$d_2$ (eq. (A8))</td>
</tr>
<tr>
<td>DALDS(MM)</td>
<td>OUTPUT</td>
<td></td>
<td>$\partial \alpha/\partial s$ at mesh points along horizontal mesh lines, $\text{rad/m}$</td>
</tr>
<tr>
<td>DALDT(MM, MHTP1)</td>
<td>OUTPUT</td>
<td></td>
<td>$\partial \alpha/\partial t$ at orthogonal mesh points, $\text{rad/m}$</td>
</tr>
<tr>
<td>DALVER(MHTP1)</td>
<td>OUTPUT</td>
<td></td>
<td>$\partial \alpha/\partial t$ at mesh points along vertical mesh lines, $\text{rad/m}$</td>
</tr>
<tr>
<td>DBL</td>
<td>TSONIN</td>
<td></td>
<td>one-half of tangential blade thickness (in radians) at intersection of a stream-line with blade leading or trailing edge</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DCHANG</td>
<td></td>
<td>COEF</td>
<td>maximum value of change in estimated values of (\partial (rV_\theta)/\partial r) at a mesh point between any two outer iterations, m/sec</td>
</tr>
<tr>
<td>DEGRAD</td>
<td></td>
<td>OUTPUT INDEV</td>
<td>conversion constant from radians to degrees</td>
</tr>
<tr>
<td>DEL</td>
<td></td>
<td>INPLOT</td>
<td>increment between plotted stream function or radius points</td>
</tr>
<tr>
<td>DELCH</td>
<td></td>
<td>OUTPUT</td>
<td>1 percent of average meridional chord length of blade, m</td>
</tr>
<tr>
<td>DELM</td>
<td></td>
<td>TSONIN</td>
<td>increment of meridional distance, m</td>
</tr>
<tr>
<td>DELMAX</td>
<td></td>
<td>VBDRY TVELCY</td>
<td>increment for (W_{hub}) at each iteration to satisfy continuity, m/sec</td>
</tr>
<tr>
<td>DELR</td>
<td></td>
<td>MESHO PTBDRY OUTPUT</td>
<td>increment in (r)-coordinate, m</td>
</tr>
<tr>
<td>DELRHO(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>difference in density, between suction and pressure surfaces, at orthogonal mesh points, kg/m(^3)</td>
</tr>
<tr>
<td>DELRNG</td>
<td></td>
<td>INPLOT</td>
<td>maximum range of points on a plot in either bottom-top or left-right directions</td>
</tr>
<tr>
<td>DELRTH</td>
<td></td>
<td>INPLOT</td>
<td>plotted blade spacing of a blade section, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>DELT</td>
<td>INPLOT</td>
<td></td>
<td>tangential blade thickness, m</td>
</tr>
<tr>
<td>DELZ</td>
<td>MESHOP</td>
<td></td>
<td>increment in z-coordinate, m</td>
</tr>
<tr>
<td>DENS</td>
<td>TVELCY</td>
<td></td>
<td>( \rho_i ) or ( \rho_o ), kg/m(^3)</td>
</tr>
<tr>
<td>DENTOL</td>
<td>TSONIN</td>
<td></td>
<td>density tolerance (input for TSONIC, ref. 5)</td>
</tr>
<tr>
<td>DFDM(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>(-B\cos\beta\frac{d(rV_\theta)}{dm}) at orthogonal mesh points (eq. (4)), m/sec</td>
</tr>
<tr>
<td>DFDS(MM)</td>
<td>BLDVEL</td>
<td></td>
<td>(\partial(rV_\theta)/\partial s) at mesh points along horizontal mesh lines, m/sec</td>
</tr>
<tr>
<td>DFDT(MM, MHTP1)</td>
<td>BLDVEL</td>
<td></td>
<td>(\partial(rV_\theta)/\partial t) at orthogonal mesh points, m/sec</td>
</tr>
<tr>
<td>DFVERT(MHTP1)</td>
<td>BLDVEL</td>
<td></td>
<td>(\partial(rV_\theta)/\partial t) at mesh points along vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>DIDR</td>
<td>NEWRHO</td>
<td></td>
<td>(\partial I/\partial r), m/sec(^2)</td>
</tr>
<tr>
<td>DIDS(MM, MHTP1)</td>
<td>NEWRHO</td>
<td></td>
<td>(\partial I/\partial s) at orthogonal mesh points, m/sec(^2)</td>
</tr>
<tr>
<td>DIDT(MHTP1)</td>
<td>NEWRHO</td>
<td></td>
<td>(\partial I/\partial t) at mesh points along vertical mesh lines, m/sec(^2)</td>
</tr>
<tr>
<td>DIST(NPPP)</td>
<td>THIKOM</td>
<td></td>
<td>distances on meridional plane along lines connecting input blade-section points (ZBL, RBL), m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DISTLE</td>
<td>INDEV</td>
<td></td>
<td>distance along horizontal mesh line from leading edge of blade for which a blade shape correction is made for incidence, m</td>
</tr>
<tr>
<td>DISTTE</td>
<td>INDEV</td>
<td></td>
<td>distance along horizontal mesh line from trailing edge of blade for which a blade shape correction is made for deviation, m</td>
</tr>
<tr>
<td>DLAM</td>
<td>TVELCY</td>
<td></td>
<td>change in $rV_\theta$ between points on vertical mesh lines, m$^2$/sec</td>
</tr>
<tr>
<td>DLDU(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>gradients of $rV_\theta$ with respect to stream function, $d(rV_\theta)/du$, at orthogonal mesh points, m$^2$/sec (This array is only defined and used in regions outside of blade row.)</td>
</tr>
<tr>
<td>DMAX</td>
<td>COEF</td>
<td></td>
<td>maximum calculated value of $\partial(rV_\theta)/\partial r$ at any mesh point, m/sec</td>
</tr>
<tr>
<td>DMD2</td>
<td>PTBDRY</td>
<td></td>
<td>expansion distance on smaller range of a plot</td>
</tr>
<tr>
<td>DMIN</td>
<td>COEF</td>
<td></td>
<td>minimum calculated value of $\partial(rV_\theta)/\partial r$ at any mesh point, m/sec</td>
</tr>
<tr>
<td>DNEW</td>
<td>INPUTT</td>
<td></td>
<td>input damping factor on calculation of $\partial(rV_\theta)/\partial r$ within blade row from outer iteration to outer iteration</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DPDR</td>
<td></td>
<td>NEWRHO</td>
<td>$\partial p''/\partial r$, N/m$^3$</td>
</tr>
<tr>
<td>DPDS(MM, MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>$\partial p''/\partial s$ at orthogonal mesh points, N/m$^3$</td>
</tr>
<tr>
<td>DPDT(MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>$\partial p''/\partial t$ at mesh points along vertical mesh lines, N/m$^3$</td>
</tr>
<tr>
<td>DPREL</td>
<td></td>
<td>TVELCY</td>
<td>change in $p''$ between points on vertical mesh lines, N/m$^2$</td>
</tr>
<tr>
<td>DRBL</td>
<td></td>
<td>TSONIN</td>
<td>tangential blade thickness at intersection of a meridional streamline with blade leading or trailing edge, m</td>
</tr>
<tr>
<td>DRBTH</td>
<td></td>
<td>THIKOM</td>
<td>interpolated value of tangential blade thickness, m</td>
</tr>
<tr>
<td>DRTHBL(NPPP, NBLPL)</td>
<td></td>
<td>THIKOM</td>
<td>tangential blade thickness at ZBL, RBL input points, m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSONIN</td>
<td></td>
</tr>
<tr>
<td>DTDRLE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta/\partial r)_le$, rad/m</td>
</tr>
<tr>
<td>DTDROM</td>
<td>THETOM</td>
<td></td>
<td>$\partial \theta/\partial r$ on orthogonal mesh, rad/m</td>
</tr>
<tr>
<td>DTDRTE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta/\partial r)_te$, rad/m</td>
</tr>
<tr>
<td>DTDS(NPPP)</td>
<td>THIKOM</td>
<td></td>
<td>$\partial \theta/\partial s$, rad/m</td>
</tr>
<tr>
<td></td>
<td>TSONIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTDSFL</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta/\partial s)_{bf}$ at leading or trailing edge, rad/m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>DTDSLE(MHTP1)</td>
<td>INDEV</td>
<td></td>
<td>$\partial \theta / \partial s$ of mid-channel flow surface at points where horizontal mesh lines cross leading edge of blade, rad/m</td>
</tr>
<tr>
<td>DTDSLE(MHTP1)</td>
<td>INDEV</td>
<td></td>
<td>$\partial \theta / \partial s$ of mid-channel flow surface at points where horizontal mesh lines cross trailing edge of blade, rad/m</td>
</tr>
<tr>
<td>DTDTLE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta / \partial t)_e$, rad/m</td>
</tr>
<tr>
<td>DTDTTE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta / \partial t)_e$, rad/m</td>
</tr>
<tr>
<td>DTDZLE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta / \partial z)_e$, rad/m</td>
</tr>
<tr>
<td>DTDZOM</td>
<td>THETOM</td>
<td></td>
<td>$\partial \theta / \partial z$ on orthogonal mesh, rad/m</td>
</tr>
<tr>
<td>DTDZTE</td>
<td>INDEV</td>
<td></td>
<td>$(\partial \theta / \partial z)_e$, rad/m</td>
</tr>
<tr>
<td>DTHDM(NPPP)</td>
<td>INPLOT</td>
<td></td>
<td>$\partial \theta / \partial s'$ used to estimate blade angle to calculate tangential blade thickness from TNBL, rad/m</td>
</tr>
<tr>
<td>DTHDR(11,NBLPL)</td>
<td>INDCOM</td>
<td></td>
<td>$\partial \theta / \partial r$ on alternate blade mesh (fig. 26), rad/m</td>
</tr>
<tr>
<td>DTHDS(MM,MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>$\partial \theta / \partial s$ at orthogonal mesh points, rad/m</td>
</tr>
<tr>
<td>DTHDSP(11,NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>$\partial \theta / \partial s'$ on alternate blade mesh (fig. 26), rad/m</td>
</tr>
<tr>
<td>DTHDT(MM,MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>$\partial \theta / \partial t$ at orthogonal mesh points, rad/m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DTHDTP(11, NBLPL)</td>
<td></td>
<td>THETOM</td>
<td>$\partial \theta / \partial t'$ on alternate blade mesh (fig. 26), rad/m</td>
</tr>
<tr>
<td>DTHDZ(11, NBLPL)</td>
<td></td>
<td>INDCOM</td>
<td>$\partial \theta / \partial z$ on alternate blade mesh (fig. 26), rad/m</td>
</tr>
<tr>
<td>DTIP</td>
<td></td>
<td>TVELCY</td>
<td>change in $T_i$ between points on vertical mesh lines, K</td>
</tr>
<tr>
<td>DTPP</td>
<td></td>
<td>TVELCY</td>
<td>change in $T''$ between points on vertical mesh lines, K</td>
</tr>
<tr>
<td>DTVEL(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>coefficients, $d$, of velocity-gradient equation ($(A7)$, part 1) at mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>DUDD(MM)</td>
<td></td>
<td>NEWRHO</td>
<td>$\partial u / \partial s$ along horizontal mesh lines, 1/m</td>
</tr>
<tr>
<td>DUDT(MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>$\partial u / \partial t$ at mesh points along vertical mesh lines, 1/m</td>
</tr>
<tr>
<td>DVDRT</td>
<td></td>
<td>COEF</td>
<td>updated estimate of $\partial (rV_\theta) / \partial r$ at a mesh point, m/sec</td>
</tr>
<tr>
<td>DVTHDR(MM, MHTP1)</td>
<td></td>
<td>COEF</td>
<td>$\partial (rV_\theta) / \partial r$ at orthogonal mesh points</td>
</tr>
<tr>
<td>DWMDM(MM, MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>$\partial W_m / \partial m$ at orthogonal mesh points, m/sec</td>
</tr>
<tr>
<td>DWMDS(MM)</td>
<td></td>
<td>TVELCY</td>
<td>$\partial W_m / \partial s$ along horizontal mesh lines, 1/sec</td>
</tr>
<tr>
<td>DWMDT(MM, MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>$\partial W_m / \partial t$ at orthogonal mesh points, 1/sec</td>
</tr>
<tr>
<td>DWMVER(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>$\partial W_m / \partial t$ along vertical mesh lines, 1/sec</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>DWTDM(MM,MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>( \frac{dW_\theta}{dm} ) at orthogonal mesh points, 1/sec</td>
</tr>
<tr>
<td>DWTDS(MM)</td>
<td></td>
<td>TVELCY</td>
<td>( \frac{\partial W_\theta}{\partial s} ) along horizontal mesh lines, 1/sec</td>
</tr>
<tr>
<td>DWTDT(MM,MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>( \frac{\partial W_\theta}{\partial t} ) at orthogonal mesh points, 1/sec</td>
</tr>
<tr>
<td>DWTVER(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>( \frac{\partial W_\theta}{\partial t} ) at mesh points along vertical mesh lines, 1/sec</td>
</tr>
<tr>
<td>DYDX(10)</td>
<td></td>
<td>INPUT</td>
<td>temporary storage for gradients</td>
</tr>
<tr>
<td>DYDX(MM or MHTP1)</td>
<td></td>
<td>PRECAL</td>
<td>temporary storage for derivative of several SPLINE and SPLINT calls</td>
</tr>
<tr>
<td>DYDX(NBLPL)</td>
<td></td>
<td>THETOM</td>
<td>temporary storage for derivative of several SPLINE calls</td>
</tr>
<tr>
<td>DYDX2(MM or MHTP1)</td>
<td></td>
<td>PRECAL</td>
<td>temporary storage for second derivatives calculated by SPLINE calls</td>
</tr>
<tr>
<td>DYDX2(NBLPL)</td>
<td></td>
<td>THETOM</td>
<td>temporary storage for second derivatives of several SPLINE calls</td>
</tr>
<tr>
<td>EM(NIN or NOUT)</td>
<td></td>
<td>TIPF</td>
<td>second derivatives of spline-fit curves</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EOP</td>
<td>INPLOT</td>
<td>MEPLOT</td>
<td>end of plot indicator (EOP = 1.0)</td>
</tr>
<tr>
<td></td>
<td>SLPLOT</td>
<td>SVPLOT</td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td>SOR</td>
<td></td>
<td>maximum absolute value of change in ( u ) at any point for an overrelaxation iteration</td>
</tr>
<tr>
<td>ERSOR(integer variable)</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>ETVEL(MHTP1)</td>
<td>TVELCY</td>
<td></td>
<td>coefficients, ( e ), of velocity-gradient equation ((A7), part I) at mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>EXPOH</td>
<td>CALCON</td>
<td></td>
<td>( 1/(\gamma - 1) )</td>
</tr>
<tr>
<td>EXTRAP</td>
<td>INDEV</td>
<td></td>
<td>distance along horizontal mesh line from blade leading or trailing edge to first mesh point outside of blade, m</td>
</tr>
<tr>
<td>FCHANG</td>
<td>BLDVEL</td>
<td></td>
<td>maximum value of change in ( F_r ) at any mesh point between any two outer iterations</td>
</tr>
<tr>
<td>FLFR(NSL)</td>
<td>INPUTT</td>
<td></td>
<td>input values of stream function designating streamlines along which output is to be printed</td>
</tr>
<tr>
<td>FMAX</td>
<td>BLDVEL</td>
<td></td>
<td>maximum new predicted value of ( F_r ) at any mesh point during an outer iteration</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>FMIN</td>
<td></td>
<td>BLDVEL</td>
<td>minimum new predicted value of $F_r$ at any mesh point during an outer iteration</td>
</tr>
<tr>
<td>FNEW</td>
<td>INPUTT</td>
<td></td>
<td>input damping factor on calculation of $F_r$ from outer iteration to outer iteration</td>
</tr>
<tr>
<td>FR(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$F_r$ at orthogonal mesh points (eq. (A4)), m/sec^2</td>
</tr>
<tr>
<td>FRAC</td>
<td></td>
<td>VBDRY</td>
<td>stream function, $u$, at mesh point on vertical boundary</td>
</tr>
<tr>
<td>FRT</td>
<td></td>
<td>BLDVEL</td>
<td>predicted value of $F_r$ at a mesh point</td>
</tr>
<tr>
<td>FST(MM, MHTP1)</td>
<td></td>
<td>BLDVEL</td>
<td>$rv_\theta$ at orthogonal mesh points, m^2/sec</td>
</tr>
<tr>
<td>FTVEL(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>coefficients, $f$, of velocity-gradient equation (A7), part I at mesh points along vertical mesh lines</td>
</tr>
<tr>
<td>FVERT(MHTP1)</td>
<td></td>
<td>BLDVEL</td>
<td>temporary storage for values of $rv_\theta$ from FST array on vertical mesh lines, m^2/sec</td>
</tr>
<tr>
<td>GAM</td>
<td>INPUTT</td>
<td></td>
<td>input, $\gamma$</td>
</tr>
<tr>
<td>GRAD(101)</td>
<td>INPLOT</td>
<td></td>
<td>dummy array for derivatives calculated by SPLINT calls in INPLOT</td>
</tr>
<tr>
<td>GRAD(MHTP1)</td>
<td>LOSSOM</td>
<td></td>
<td>dummy array of derivatives calculated in SPLINT call</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>H1</td>
<td>COEF</td>
<td></td>
<td>$h_1$ (eq. (A8), fig. 24)</td>
</tr>
<tr>
<td>H2</td>
<td>COEF</td>
<td></td>
<td>$h_2$ (eq. (A8), fig. 24)</td>
</tr>
<tr>
<td>H3</td>
<td>COEF</td>
<td></td>
<td>$h_3$ (eq. (A8), fig. 24)</td>
</tr>
<tr>
<td>H4</td>
<td>COEF</td>
<td></td>
<td>$h_4$ (eq. (A8), fig. 24)</td>
</tr>
<tr>
<td>IDEBUG</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating multiple of outer iterations at which debug output is printed</td>
</tr>
<tr>
<td>IEND</td>
<td>Blank</td>
<td></td>
<td>integer indicator of stage of solution to which program has proceeded:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$IEND = -1$, prior to convergence of subsonic solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$IEND = 0$, between convergence of subsonic solution and beginning of transonic solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$IEND = 1$, during first transonic solution with all velocities smaller than choking-mass-flow solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$IEND = 2$, during second transonic solution with all velocities greater than choking-mass-flow solution</td>
</tr>
<tr>
<td>IL</td>
<td>VBDRY</td>
<td></td>
<td>integer (1 or 3) to identify proper word in BDY array</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>ILE(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>vertical mesh line numbers of first mesh point inside blade region at leading edge</td>
</tr>
<tr>
<td>ILS(NSL)</td>
<td>SLCOM</td>
<td></td>
<td>vertical mesh line number of first intersection of a streamline with a vertical mesh line inside blade region at leading edge</td>
</tr>
<tr>
<td>IMESH</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating the multiple of outer iterations at which major output is printed for orthogonal mesh</td>
</tr>
<tr>
<td>IND</td>
<td>VBDRY, TVELCY</td>
<td></td>
<td>integer which indicates solution procedure in CONTIN</td>
</tr>
<tr>
<td>INTVL</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>ILOT</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating multiple of outer iterations at which major output is plotted on microfilm</td>
</tr>
<tr>
<td>IPLT</td>
<td>MEPLOT</td>
<td></td>
<td>indicates which of two plots is being made</td>
</tr>
<tr>
<td>ISLINE</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating multiple of outer iterations at which major output is printed along streamlines</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>ISTATL</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating multiple of outer iterations at which major output is printed along station lines</td>
</tr>
<tr>
<td>ISUPER</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating whether only subsonic, or both subsonic and supersonic, solutions of velocity-gradient equation are to be calculated</td>
</tr>
<tr>
<td>ITE(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>vertical mesh line numbers of last mesh point inside blade region at trailing edge</td>
</tr>
<tr>
<td>ITER</td>
<td>Blank</td>
<td></td>
<td>outer iteration counter, incremented by 1 at beginning of each outer iteration</td>
</tr>
<tr>
<td>ITS(NSL)</td>
<td>SLCOM</td>
<td></td>
<td>vertical mesh line number of last intersection of a streamline with a vertical mesh line inside blade region at trailing edge</td>
</tr>
<tr>
<td>ITSON</td>
<td>INPUTT</td>
<td></td>
<td>integer input indicating multiple of outer iterations at which information is printed as input for TSONIC program (ref. 5)</td>
</tr>
<tr>
<td>JZ</td>
<td>VBDRY</td>
<td>TVELCY</td>
<td>integer used to indicate to CONTIN that subsonic (JZ = 1) or supersonic (JZ = 2) solution is desired</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>K(MM, MHTP1) (real variable)</td>
<td>VARCOM</td>
<td></td>
<td>(k_0) (eq. (A9)) at orthogonal mesh points</td>
</tr>
<tr>
<td>KNEW(real variable)</td>
<td>COEF</td>
<td></td>
<td>updated value of (k_0) (eq. (A9)) at a mesh point</td>
</tr>
<tr>
<td>LAMBDA(MHTP1) (real variable)</td>
<td>VBDRY</td>
<td></td>
<td>(\lambda) for mesh points along vertical mesh lines, (m^2/sec)</td>
</tr>
<tr>
<td>LAMBDOMHTP1(real variable)</td>
<td>TVELCY</td>
<td></td>
<td>((rV_\theta)) for mesh points along vertical mesh lines, (m^2/sec)</td>
</tr>
<tr>
<td>LAMDAI(real variable)</td>
<td>PRECAL</td>
<td></td>
<td>(\lambda), (m^2/sec)</td>
</tr>
<tr>
<td>LAMIN(NIN)</td>
<td>INPUTT</td>
<td></td>
<td>input values of (\lambda) at points along line from hub to shroud on which upstream flow conditions are given, (m^2/sec)</td>
</tr>
<tr>
<td>LAMOUT(NOUT)</td>
<td>INPUTT</td>
<td></td>
<td>input values of ((rV_\theta)) at points along line from hub to shroud on which downstream flow conditions are given, (m^2/sec)</td>
</tr>
<tr>
<td>LAMVT</td>
<td>INPUTT</td>
<td></td>
<td>input integer (0 or 1) indicating whether upstream and downstream whirl (0) or tangential velocity (1) is given as input</td>
</tr>
<tr>
<td>LMAX(real variable)</td>
<td>SOR</td>
<td></td>
<td>maximum value of (\text{RATIO}) over all mesh points</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LMIN(real variable)</td>
<td>SOR</td>
<td></td>
<td>minimum value of RATIO over all mesh points</td>
</tr>
<tr>
<td>LOC</td>
<td>VBDRY</td>
<td></td>
<td>integer (1 or MM) indicating vertical mesh line number for which VBDRY is called</td>
</tr>
<tr>
<td>LOSOUT(NOUT)</td>
<td>INPUTT</td>
<td></td>
<td>input fraction of absolute total pressure loss, at points along line from hub to shroud on which downstream flow conditions are given</td>
</tr>
<tr>
<td>LRNG(real variable)</td>
<td>INPLOT</td>
<td>SVPLOT</td>
<td>left-most point of range of a plot</td>
</tr>
<tr>
<td>LSFR</td>
<td>INPUTT</td>
<td></td>
<td>input integer (0 or 1) indicating whether upstream and downstream flow conditions are input as a function of stream function (0) or radius (1)</td>
</tr>
<tr>
<td>LTPL</td>
<td>INPUTT</td>
<td></td>
<td>input integer (0 or 1) indicating whether downstream total pressure (0) or fractional loss of stagnation pressure (1) is given in input</td>
</tr>
<tr>
<td>MARK</td>
<td>CROSCM</td>
<td></td>
<td>marker in MESHO and CROSCD to indicate if CROSCD is being called for the first time</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>MARK(NOSTAT)</td>
<td>OUTPUT</td>
<td></td>
<td>integers between 1 and 4 indicating whether output station lines are outside blade, within blade, or on leading or trailing edge</td>
</tr>
<tr>
<td>MBI</td>
<td>INPUTT</td>
<td></td>
<td>input number of vertical mesh lines from left boundary of orthogonal mesh (ZOMIN) to first point of mesh-size change (ZOMBI)</td>
</tr>
<tr>
<td>MBL(NPPP,NBLPL)</td>
<td>INPLOT</td>
<td></td>
<td>s'-coordinate, corresponding to ZBL and RBL, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>MBLD</td>
<td>SVPLOT</td>
<td></td>
<td>number of suction-surface or pressure-surface velocities on a plot</td>
</tr>
<tr>
<td>MBO</td>
<td>INPUTT</td>
<td></td>
<td>input total number of vertical mesh lines from left boundary of orthogonal mesh (ZOMIN) to point of second mesh-size change (ZOMBO)</td>
</tr>
<tr>
<td>MCT</td>
<td>OUTPUT</td>
<td></td>
<td>integer (1 or 2) indicating whether a compressor (1) or a turbine (2) is being analyzed</td>
</tr>
<tr>
<td>MHT</td>
<td>INPUTT</td>
<td></td>
<td>input total number of horizontal mesh spaces from hub to shroud of orthogonal mesh; maximum of 100</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>MHTP1</td>
<td>CALCON</td>
<td>TSONIN</td>
<td>MHT + 1</td>
</tr>
<tr>
<td>MLESP(5)</td>
<td></td>
<td>TSONIN</td>
<td>m-coordinates of blade surface spline points near leading edge of a blade section, printed to make a layout to obtain input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MM</td>
<td>INPUTT</td>
<td></td>
<td>input total number of vertical mesh lines from left to right boundaries of orthogonal mesh (ZOMIN to ZOMOUT), maximum of 100</td>
</tr>
<tr>
<td>MMM1</td>
<td>CALCON</td>
<td></td>
<td>MM - 1</td>
</tr>
<tr>
<td>MR(MM)</td>
<td>TSONIN</td>
<td></td>
<td>m-coordinates of points defining a stream channel, printed as input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>MSFL</td>
<td>INPUTT</td>
<td></td>
<td>input total mass flow through entire circumferential annulus of machine, kg/sec</td>
</tr>
<tr>
<td>MSL(MM,NSL)</td>
<td>SLCOM</td>
<td></td>
<td>m-coordinates of points along streamlines where they cross vertical mesh lines, m (Origin of m-coordinate along a streamline corresponds to point where ( z = 0 ) along streamline.)</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>MSP(MM)(real variable)</td>
<td></td>
<td>TSONIN</td>
<td>m-coordinates of blade surface spline points, given as input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>MST(NSL,NOSTAT),</td>
<td></td>
<td>STACOM</td>
<td>m-coordinates of points where station lines cross streamlines, m (Origin of m-coordinates along a streamline corresponds to point where z = 0 along streamline.)</td>
</tr>
<tr>
<td>(real variable)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTEM(NOSTAT)</td>
<td></td>
<td>OUTPUT</td>
<td>temporary storage for values of m-coordinate for MST array, m</td>
</tr>
<tr>
<td>(real variable)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTESP(5)</td>
<td></td>
<td>TSONIN</td>
<td>m-coordinates of blade surface spline points near trailing edge of a blade section, printed to make a layout to obtain input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>(real variable)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NBL</td>
<td></td>
<td>INPUTT</td>
<td>input number of blades in blade row</td>
</tr>
<tr>
<td>NBLPL</td>
<td></td>
<td>INPUTT</td>
<td>number of input blade planes or blade sections on which data (ZBL, RBL, THBL, TNBL) are given to describe mean flow surface and blade thickness, maximum of 50</td>
</tr>
<tr>
<td>NBLPTS</td>
<td></td>
<td>TSONIN</td>
<td>number of spline points on suction or pressure surface of a blade section</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>NCOUNT</td>
<td>VBDRY</td>
<td>TVELCY</td>
<td>total number of iterations or attempts at satisfying velocity-gradient equation</td>
</tr>
<tr>
<td>NHUB</td>
<td>INPUTT</td>
<td></td>
<td>number of input data points in ZHUB and RHUB arrays, maximum of 50</td>
</tr>
<tr>
<td>NIN</td>
<td>INPUTT</td>
<td></td>
<td>number of input data points in upstream arrays of flow properties (SFIN, RADIN, TIP, PRIP, LAMIN, VTHIN), maximum of 50</td>
</tr>
<tr>
<td>NOSTAT</td>
<td>INPUTT</td>
<td></td>
<td>input number of hub-shroud stations (located by coordinates in ZHST and ZTST) at which output is desired, maximum of 50</td>
</tr>
<tr>
<td>NOUT</td>
<td>INPUTT</td>
<td></td>
<td>number of input data points in downstream arrays of flow properties (SFOUT, RADOUT, PROP, LOSOUT, LAMOUT, VTHOUT), maximum of 50</td>
</tr>
<tr>
<td>NPPP</td>
<td>INPUTT</td>
<td></td>
<td>number of input data points per blade section or blade plane in ZBL, RBL, THBL, and TNBL arrays, maximum of 50</td>
</tr>
<tr>
<td>NREAD</td>
<td>Blank</td>
<td></td>
<td>integer number of input tape-reading unit of computer which is running MERIDL</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>NRSP</td>
<td>INPUTTT</td>
<td>TSONIN</td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>NSL</td>
<td>INPUTTT</td>
<td></td>
<td>input number of streamlines from hub to shroud (designated by values in FLFR) at which output is desired, maximum of 50</td>
</tr>
<tr>
<td>NTIP</td>
<td>INPUTTT</td>
<td></td>
<td>number of input data points in ZTIP and RTIP arrays, maximum of 50</td>
</tr>
<tr>
<td>NWRT</td>
<td>Blank</td>
<td></td>
<td>integer number of output tape-writing unit of computer which is running MERIDL</td>
</tr>
<tr>
<td>OMEGA</td>
<td>INPUTTT</td>
<td></td>
<td>input rotational speed, $\omega$, rad/sec</td>
</tr>
<tr>
<td>ORF</td>
<td>SOR</td>
<td></td>
<td>overrelaxation factor</td>
</tr>
<tr>
<td>ORF</td>
<td>TSONIN</td>
<td></td>
<td>overrelaxation factor (input for TSONIC, ref. 5)</td>
</tr>
<tr>
<td>ORFMAX</td>
<td>SOR</td>
<td></td>
<td>current estimate for maximum value of ORF calculated using LMAX</td>
</tr>
<tr>
<td>ORFMIN</td>
<td>SOR</td>
<td></td>
<td>current estimate for minimum value of ORF calculated using LMIN</td>
</tr>
<tr>
<td>PHI</td>
<td>OUTPUT</td>
<td></td>
<td>$\phi$, deg</td>
</tr>
<tr>
<td>PITCH</td>
<td>CALCON</td>
<td></td>
<td>$2\pi/NBL$, rad</td>
</tr>
<tr>
<td>PLOSS(MM,MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>fractional loss of relative total pressure at orthogonal mesh points</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>PLOSSL</td>
<td></td>
<td>TSONIN</td>
<td>fractional loss of relative total pressure at a ZSL,RSL point along a streamline</td>
</tr>
<tr>
<td>PLOSTE</td>
<td></td>
<td>INDEV</td>
<td>fractional loss of relative total pressure at blade trailing edge</td>
</tr>
<tr>
<td>PLTX(101)</td>
<td></td>
<td>INPLOT</td>
<td>temporary storage of x-plot coordinates of many arrays in INPLOT</td>
</tr>
<tr>
<td>PLTY(101)</td>
<td></td>
<td>INPLOT</td>
<td>temporary storage of y-plot coordinates of many arrays in INPLOT</td>
</tr>
<tr>
<td>PPTHBL(NPPP,NBLPL)</td>
<td></td>
<td>INPLOT</td>
<td>values of $r\theta$ for pressure surface of adjacent blade, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>PRATIO(MHTP1)</td>
<td></td>
<td>LOSSOM</td>
<td>$p''/\left(p''<em>{0}\right)</em>{\text{ideal}}$ for each horizontal mesh line downstream of blade</td>
</tr>
<tr>
<td>PREL(MM or MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>$p''$ at mesh points along horizontal or vertical mesh lines, N/m$^2$</td>
</tr>
<tr>
<td>PRELN</td>
<td></td>
<td>TVELCY</td>
<td>new $p''$, N/m$^2$</td>
</tr>
<tr>
<td>PRINP</td>
<td></td>
<td>PRECAL</td>
<td>$p'_1$, N/m$^2$</td>
</tr>
<tr>
<td>PRIP(NIN)</td>
<td></td>
<td>INPUTT</td>
<td>input, $p'_1$, at points along line from hub to shroud on which upstream flow conditions are given, N/m$^2$</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>PROP(NOUT)</td>
<td>INPUTT</td>
<td></td>
<td>input, pO, at points along line from hub to shroud on which downstream flow conditions are given, N/m²</td>
</tr>
<tr>
<td>PTHBL(NPPP,NBLPL)</td>
<td>INPLOT</td>
<td></td>
<td>values of rθ for pressure surface, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>R1</td>
<td>MESHO</td>
<td></td>
<td>r-coordinate of intersection of line (1), fig. 8, with upper horizontal mesh line, m</td>
</tr>
<tr>
<td>R2</td>
<td>MESHO</td>
<td></td>
<td>r-coordinate of intersection of line (2), fig. 8, with upper horizontal mesh line, m</td>
</tr>
<tr>
<td>RADIN(NIN)</td>
<td>INPUTT</td>
<td></td>
<td>input r-coordinates of points along line from hub to shroud on which upstream flow conditions are given, m</td>
</tr>
<tr>
<td>RADOUT(NOUT)</td>
<td>INPUTT</td>
<td></td>
<td>input r-coordinates of points along line from hub to shroud on which downstream flow conditions are given, m</td>
</tr>
<tr>
<td>RATIO</td>
<td>SOR</td>
<td></td>
<td>uᵢ^{m+1}/uᵢ^m for use in eqs. (B2) and (B3) of ref. 11</td>
</tr>
<tr>
<td>RBL(NPPP,NBLPL)</td>
<td>INPUTT</td>
<td></td>
<td>input array of r-coordinates, corresponding to ZBL, of points describing mean blade surface, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON Subroutine</td>
<td>Description and comments</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td></td>
</tr>
<tr>
<td>RBLTEM(NBLPL)</td>
<td>OUTPUT</td>
<td>temporary storage for values in RLE and RTE arrays, m</td>
<td></td>
</tr>
<tr>
<td>RBRNG</td>
<td>PLTCOM</td>
<td>r-coordinate of bottom boundary of a plot of meridional plane or orthogonal mesh, done in MEPLOT, m</td>
<td></td>
</tr>
<tr>
<td>RCARB(MHTP1)</td>
<td>TVELCY</td>
<td>$\rho \cos(\alpha - \varphi) rB$ along a vertical mesh line (eq. (5)), kg/m$^2$</td>
<td></td>
</tr>
<tr>
<td>RCURV</td>
<td>CROSCD</td>
<td>r-coordinate of horizontal mesh line at input z-coordinate, m</td>
<td></td>
</tr>
<tr>
<td>REDFAC</td>
<td>INPUTT</td>
<td>input factor used to reduce mass flow (MSFL) in order to assure subsonic flow throughout flow passage</td>
<td></td>
</tr>
<tr>
<td>REFR</td>
<td>CROSCM</td>
<td>reference r-coordinate in MESHO from which orthogonal mesh is extended by addition of another &quot;link,&quot; m</td>
<td></td>
</tr>
<tr>
<td>REFSL</td>
<td>CROSCM</td>
<td>reference slope in MESHO of vertical link being extended from a known orthogonal mesh point to a new mesh point</td>
<td></td>
</tr>
<tr>
<td>REFZ</td>
<td>CROSCM</td>
<td>reference z-coordinate in MESHO from which orthogonal mesh is extended by addition of another &quot;link,&quot; m</td>
<td></td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>RELER</td>
<td></td>
<td>NEWRHO</td>
<td>maximum relative change in $W$ at any mesh point between two outer iterations</td>
</tr>
<tr>
<td>RELTOP(NOUT)</td>
<td></td>
<td>PRECAL</td>
<td>$p_i''/p_o''$ at hub-shroud input points downstream of blade row</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOSSOM</td>
<td></td>
</tr>
<tr>
<td>REPEAT</td>
<td></td>
<td>VBDRY</td>
<td>logical variable indicating that velocity-gradient solution should be repeated with new values of TIPBDY, RHOIP, and LAMBDA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TVELCY</td>
<td></td>
</tr>
<tr>
<td>RHIN</td>
<td></td>
<td>CALCON</td>
<td>$r$-coordinate of intersection with hub profile of line on which upstream flow conditions are given, m</td>
</tr>
<tr>
<td>RHO(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>$\rho$, at orthogonal mesh points, kg/m$^3$</td>
</tr>
<tr>
<td>RHOAV(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>average density across flow channel from suction surface to pressure surface, at orthogonal mesh points, kg/m$^3$</td>
</tr>
<tr>
<td>RHOIP(MHTP1)</td>
<td></td>
<td>VBDRY</td>
<td>$\rho_i'$ for mesh points along vertical mesh lines, kg/m$^3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TVELCY</td>
<td></td>
</tr>
<tr>
<td>RHOIP(NIN)</td>
<td></td>
<td>RHOIPF</td>
<td>$\rho_i'$ at input points of upstream flow conditions, kg/m$^3$</td>
</tr>
<tr>
<td>RHOIP</td>
<td></td>
<td>INIT</td>
<td>$\rho_i'$ at hub, kg/m$^3$</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>RHOIP</td>
<td>TSONIN</td>
<td>$\rho_i'$ for a streamline, kg/m$^3$, input for TSONIC (ref. 5)</td>
<td></td>
</tr>
<tr>
<td>RHOL</td>
<td>BLDVEL</td>
<td>$\rho_l$, kg/m$^3$</td>
<td></td>
</tr>
<tr>
<td>RHOOP(MHTP1)</td>
<td>TVELCY</td>
<td>$\rho_o'$ for mesh points along vertical mesh lines, kg/m$^3$</td>
<td></td>
</tr>
<tr>
<td>RHOOP(NOUT)</td>
<td>RHOOPF</td>
<td>$\rho_i'$ at input points of downstream flow conditions, kg/m$^3$</td>
<td></td>
</tr>
<tr>
<td>RHOSL</td>
<td>TSONIN</td>
<td>$\rho$ at a ZSL, RSL point along a streamline, kg/m$^3$</td>
<td></td>
</tr>
<tr>
<td>RHT</td>
<td>BLDVEL</td>
<td>$\rho_{tr}$, kg/m$^3$</td>
<td></td>
</tr>
<tr>
<td>RHOUT</td>
<td>CALCON</td>
<td>r-coordinate of intersection with hub profile of line on which downstream flow conditions are given, m</td>
<td></td>
</tr>
<tr>
<td>RHOW</td>
<td>VBDRY</td>
<td>$\rho_W$, kg/(m$^2$)(sec)</td>
<td></td>
</tr>
<tr>
<td>RHPLT(100)</td>
<td>PLTCOM</td>
<td>r-coordinates used for plotting hub profile, RHUB, in MEPLOT, m</td>
<td></td>
</tr>
<tr>
<td>RHUB(NHUB)</td>
<td>INPUTT</td>
<td>input r-coordinates of points defining hub or bottom boundary of flow channel, m</td>
<td></td>
</tr>
<tr>
<td>RILOM(MHTP1)</td>
<td>LAMDAF</td>
<td>radii for spline fit of stream function against radius</td>
<td></td>
</tr>
<tr>
<td>RLE(NBLPL)</td>
<td>CALCON</td>
<td>r-coordinates of input blade section points (from RBL) defining leading edge of blade, m</td>
<td></td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>RLEH</td>
<td>CALCON</td>
<td></td>
<td>$r$-coordinate of intersection of leading edge of blade with hub profile, m</td>
</tr>
<tr>
<td>RLEOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>$r$-coordinates of intersections of horizontal mesh lines with blade leading edge, m</td>
</tr>
<tr>
<td>RLEP</td>
<td>TSONIN</td>
<td></td>
<td>$r$-coordinate of point near leading edge of blade section along meridional streamline, m</td>
</tr>
<tr>
<td>RLES SL</td>
<td>TSONIN</td>
<td></td>
<td>$r$-coordinate of intersection of a streamline with leading edge of blade, m</td>
</tr>
<tr>
<td>RLET</td>
<td>CALCON</td>
<td></td>
<td>$r$-coordinate of intersection of leading edge of blade with shroud profile, m</td>
</tr>
<tr>
<td>RLINE</td>
<td>CROSCD</td>
<td></td>
<td>$r$-coordinate on straight-line vertical orthogonal link at input z-coordinate, m</td>
</tr>
<tr>
<td>RLP LT(100)</td>
<td>PLTCOM</td>
<td></td>
<td>$r$-coordinates used for plotting blade leading edge, RLE, in ME PLOT, m</td>
</tr>
<tr>
<td>RMEAN</td>
<td>VBDRY</td>
<td></td>
<td>mean radius $(r_{hub} + r_{tip})/2$, m</td>
</tr>
<tr>
<td>RMR</td>
<td>MESHO, CROSCD</td>
<td></td>
<td>$RCURV - RLINE$ in CROSCD, m</td>
</tr>
<tr>
<td>RM SP(MM)</td>
<td>TSONIN</td>
<td></td>
<td>$r$-coordinates of points defining a stream channel, printed as input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>RLOM(MHTP1)</td>
<td></td>
<td>RVTHTA</td>
<td>radii for spline fit of stream function against radius</td>
</tr>
<tr>
<td>ROM(MM, MHTP1)</td>
<td></td>
<td>CALCON</td>
<td>r-coordinates of orthogonal mesh, m</td>
</tr>
<tr>
<td>ROTI(MM or MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>rothalpy, I, at mesh points along horizontal or vertical mesh lines, m^2/sec^2</td>
</tr>
<tr>
<td>RPC(11, NBLPL)</td>
<td></td>
<td>INDCOM</td>
<td>r-coordinates of alternate mesh (fig. 26)</td>
</tr>
<tr>
<td>RRAD(NHUB, MHTP1)</td>
<td></td>
<td>MESHO</td>
<td>r-coordinates of points along radial lines from input points on hub profile to shroud profile, m</td>
</tr>
<tr>
<td>RRNG</td>
<td></td>
<td>INPLOT</td>
<td>right-most point of range of a plot</td>
</tr>
<tr>
<td>RRTHBL(NPPP, NBLPL)</td>
<td></td>
<td>INPLOT</td>
<td>values of ( r \theta ) coordinate of blade mean camber line of adjacent blade, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>RSL(MM, NSL)</td>
<td></td>
<td>SLCOM</td>
<td>array of r-coordinates of points along streamlines where they cross vertical mesh lines, m</td>
</tr>
<tr>
<td>RSLTEM(NSL)</td>
<td></td>
<td>OUTPUT</td>
<td>temporary storage for calculated values to be put into RSL array, m</td>
</tr>
<tr>
<td>RSPLT(100)</td>
<td></td>
<td>PLTCOM</td>
<td>r-coordinates used for plotting shroud profile, RTIP, in MEPLLOT, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RST(NSL,NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>r-coordinates of points along station lines where they cross streamlines, m</td>
</tr>
<tr>
<td>RTE(NBLPL)</td>
<td>CALCON</td>
<td></td>
<td>r-coordinates of input blade section points (from RBL) defining trailing edge of blade, m</td>
</tr>
<tr>
<td>RTEH</td>
<td>CALCON</td>
<td></td>
<td>r-coordinate of intersection of trailing edge of blade with hub profile, m</td>
</tr>
<tr>
<td>RTEM(10)</td>
<td>INPUT</td>
<td></td>
<td>temporary storage for r-coordinates, m</td>
</tr>
<tr>
<td>RTEM(MM)</td>
<td>MEPLOT</td>
<td></td>
<td>temporary storage of r-coordinates from ROM for plotting, m</td>
</tr>
<tr>
<td>RTEM(NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>temporary storage of r-coordinates from RPC, m</td>
</tr>
<tr>
<td>RTEM(MHTP1 or NOSTAT or 20)</td>
<td>OUTPUT</td>
<td></td>
<td>temporary storage for values from ROM array on vertical mesh lines; also temporary storage for values from RST array along station lines, m</td>
</tr>
<tr>
<td>RTEOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>r-coordinates of intersections of horizontal mesh lines with blade trailing edge, m</td>
</tr>
<tr>
<td>RTEP</td>
<td>TSONIN</td>
<td></td>
<td>r-coordinate of point near trailing edge of blade section along meridional streamline, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RTESL</td>
<td></td>
<td>TSONIN</td>
<td>r-coordinate of intersection of a streamline with trailing edge of blade, m</td>
</tr>
<tr>
<td>RTET</td>
<td></td>
<td>CALCON</td>
<td>r-coordinate of intersection of trailing edge of blade with shroud profile, m</td>
</tr>
<tr>
<td>RTHBL(NPPP,NBLPL)</td>
<td></td>
<td>INPLOT</td>
<td>rθ-coordinate of blade mean camber line, corresponding to points in ZBL,RBL, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>RTIN</td>
<td></td>
<td>CALCON</td>
<td>calculated r-coordinate of intersection with shroud profile of line on which upstream flow conditions are given, m</td>
</tr>
<tr>
<td>RTIP(NTIP)</td>
<td></td>
<td>INPUTT</td>
<td>input r-coordinates of points defining shroud or top boundary of flow channel, m</td>
</tr>
<tr>
<td>RTLEP1(5)</td>
<td></td>
<td>TSONIN</td>
<td>rθ at blade suction-surface spline points near leading edge of a blade section, referenced to zero at leading edge, used to make a layout to obtain input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>RTLEP2(5)</td>
<td></td>
<td>TSONIN</td>
<td>( r \theta ) at blade pressure-surface spline points near leading edge of a blade section, referenced to zero at leading edge, used to make a layout to obtain input for TSONIC (ref. 5), m</td>
</tr>
<tr>
<td>RTMP(NHUB)</td>
<td></td>
<td>CROSCM</td>
<td>temporary storage for portions of RRAD array in MESHO and CROSCD, m</td>
</tr>
<tr>
<td>RTOLER</td>
<td></td>
<td>VBDRY</td>
<td>tolerance on relative error of subsequent calculated values of integrated mass flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TVELCY</td>
<td></td>
</tr>
<tr>
<td>RTOUT</td>
<td></td>
<td>CALCON</td>
<td>calculated ( r )-coordinate of intersection with shroud profile of line on which downstream flow conditions are given, m</td>
</tr>
<tr>
<td>RTPLT(100)</td>
<td></td>
<td>PLTCOM</td>
<td>( r )-coordinate used for plotting blade trailing edge, RTE, in MEPLOT, m</td>
</tr>
<tr>
<td>RTRNG</td>
<td></td>
<td>PLTCOM</td>
<td>( r )-coordinate of top boundary of a plot of meridional plane or orthogonal mesh, done in MEPLOT, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>RTTEP1(5)</td>
<td>TSONIN</td>
<td>rθ at blade suction-surface spline points near trailing edge of a blade section, referenced to zero at trailing edge, used to make a layout to obtain input for TSONIC (ref. 5), m</td>
<td></td>
</tr>
<tr>
<td>RTTEP2(5)</td>
<td>TSONIN</td>
<td>rθ at blade pressure-surface spline points near trailing edge of a blade section, referenced to zero at trailing edge, used to make a layout to obtain input for TSONIC (ref. 5), m</td>
<td></td>
</tr>
<tr>
<td>RVA</td>
<td>VBDRY</td>
<td>ρ_j \cdot W_j \cdot \cos \beta_j \cdot 2π r_j/NBL, kg/(m)(sec)</td>
<td></td>
</tr>
<tr>
<td>RVA</td>
<td>TVELCY</td>
<td>ρ_j \cdot W_j \cdot \cos \beta_j \cdot \cos(α - φ)_j \cdot r_j B_j, kg/(m)(sec)</td>
<td></td>
</tr>
<tr>
<td>RVAS</td>
<td>VBDRY</td>
<td>ρ_{j+1} \cdot W_{j+1} \cdot \cos \beta_{j+1} \cdot 2π r_{j+1}/NBL, kg/(m)(sec)</td>
<td></td>
</tr>
<tr>
<td>RVAS</td>
<td>TVELCY</td>
<td>ρ_{j+1} \cdot W_{j+1} \cdot \cos \beta_{j+1} \cdot \cos(α - φ)<em>{j+1} \cdot r</em>{j+1} B_{j+1}, kg/(m)(sec)</td>
<td></td>
</tr>
<tr>
<td>SAL</td>
<td>TVELCY</td>
<td>sin α</td>
<td></td>
</tr>
<tr>
<td>SAMP(MM, MHTP1)</td>
<td>VARCOM</td>
<td>sin(α - φ) at orthogonal mesh points</td>
<td></td>
</tr>
<tr>
<td>SAMPLE</td>
<td>INDEV</td>
<td>sin(α - φ)_{le}</td>
<td></td>
</tr>
<tr>
<td>SAMPTE</td>
<td>INDEV</td>
<td>sin(α - φ)_{te}</td>
<td></td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SBETA</td>
<td></td>
<td>TVELCY</td>
<td>$\sin \beta$</td>
</tr>
<tr>
<td>SDIST</td>
<td></td>
<td>INDEV</td>
<td>DISTLE (or DISTTE) plus the distance from blade leading (or trailing) edge out to first adjacent mesh point, m</td>
</tr>
<tr>
<td>SDRIV(NHUB)</td>
<td></td>
<td>CROSCM</td>
<td>second derivatives of a spline fit of horizontal rows in RRAD array in MESHO, $1/m$</td>
</tr>
<tr>
<td>SF(MHTP1)</td>
<td></td>
<td>LOSSOM</td>
<td>estimate of stream function at mesh points on down-stream boundary of orthogonal mesh</td>
</tr>
<tr>
<td>SFIN(NIN)</td>
<td></td>
<td>INPUTT</td>
<td>input values of stream function along hub-shroud line on which upstream flow conditions are given</td>
</tr>
<tr>
<td>SFOUT(NOUT)</td>
<td></td>
<td>INPUTT</td>
<td>input values of stream function along hub-shroud line on which downstream flow conditions are given</td>
</tr>
<tr>
<td>SINBTA</td>
<td></td>
<td>VBDRY</td>
<td>$\sin \beta$</td>
</tr>
<tr>
<td>SL1</td>
<td></td>
<td>MESHO</td>
<td>slope of horizontal orthogonal at its point of intersection by a radial line in CROSCD</td>
</tr>
<tr>
<td>SLCRD(integer variable)</td>
<td></td>
<td>TSONIN</td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>SLEOM(MHTP1)</td>
<td></td>
<td>CALCON</td>
<td>s-coordinates of intersections of horizontal mesh lines with blade leading edge, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>SLIDLE</td>
<td>INDEV</td>
<td></td>
<td>solidity at leading edge of blade where it is intersected by a horizontal mesh line</td>
</tr>
<tr>
<td>SLIDTE</td>
<td>INDEV</td>
<td></td>
<td>solidity at trailing edge of blade where it is intersected by a horizontal mesh line</td>
</tr>
<tr>
<td>SLOM(MM)</td>
<td>MESHO</td>
<td></td>
<td>slopes of horizontal mesh lines at mesh points</td>
</tr>
<tr>
<td>SLOPE(NIN or NOUT)</td>
<td>TIPF</td>
<td></td>
<td>derivatives of spline-fit curves</td>
</tr>
<tr>
<td></td>
<td>RHOIPF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RHOOPF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAMDAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RVTHTA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOM(MM, MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>s-coordinates of orthogonal mesh, (m)</td>
</tr>
<tr>
<td>SPHI(MM, MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>(\sin \varphi) at orthogonal mesh points</td>
</tr>
<tr>
<td>SPHILE</td>
<td>INDEV</td>
<td></td>
<td>(\sin \varphi_{le})</td>
</tr>
<tr>
<td>SPHITE</td>
<td>INDEV</td>
<td></td>
<td>(\sin \varphi_{te})</td>
</tr>
<tr>
<td>SPLNO1</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>SPLNO2</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>SRE(integer variable)</td>
<td>Blank</td>
<td></td>
<td>switch used to turn on and off printing of error or warning messages in some subroutines</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>SRW(integer variable)</td>
<td>Blank</td>
<td></td>
<td>switch used to turn on and off printing of debug information in some subroutines</td>
</tr>
<tr>
<td>STHBL(NPPP,NBLPL)</td>
<td>INPLOT</td>
<td></td>
<td>values of $r\theta$ for suction surface of adjacent blade, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>STEOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>s-coordinates of intersections of horizontal mesh lines with blade trailing edge, m</td>
</tr>
<tr>
<td>STHBL(NPPP,NBLPL)</td>
<td>INPLOT</td>
<td></td>
<td>values of $r\theta$ on suction surface, used for plotting input blade sections, m</td>
</tr>
<tr>
<td>STRFN(integer variable)</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>SURVL(integer variable)</td>
<td>TSONIN</td>
<td></td>
<td>input for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>SYM</td>
<td>INPLOT, MEPLOT, SLPLOT, SVPLOT</td>
<td></td>
<td>indicator used to select a special plot symbol from a table</td>
</tr>
<tr>
<td>SYN</td>
<td>MEPLOT</td>
<td></td>
<td>indicator used to select a special plot symbol from a table</td>
</tr>
<tr>
<td>SZRBL(NPPP)</td>
<td>THETOM</td>
<td></td>
<td>arc length along input blade section in meridional plane, m</td>
</tr>
<tr>
<td>SZRPC(11 or NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>arc length along vertical or horizontal lines of alternate mesh (fig. 26)</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TANBBL</td>
<td>INDEV</td>
<td></td>
<td>$\tan \beta_b$, tangent of blade mean camber line angle at leading or trailing edge</td>
</tr>
<tr>
<td>TANBFL</td>
<td>INDEV</td>
<td></td>
<td>$\tan \beta_{bf}$, tangent of flow angle at leading or trailing edge, corrected for block-age</td>
</tr>
<tr>
<td>TEMPER</td>
<td>TVELCY</td>
<td></td>
<td>$T_1'$ or $T_o'$, $K$</td>
</tr>
<tr>
<td>TGROG</td>
<td>CALCON</td>
<td></td>
<td>$2\gamma R/(\gamma + 1)$</td>
</tr>
<tr>
<td>THBL(NPPP,NBLPL)</td>
<td>INPUTT</td>
<td></td>
<td>input array of $\theta$-coordinates, corresponding to $RBL, ZBL$, of points describing mean blade surface, rad</td>
</tr>
<tr>
<td>THLEOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>$\theta$-coordinates of intersections of horizontal mesh lines with blade leading edge, rad</td>
</tr>
<tr>
<td>THLESL</td>
<td>TSONIN</td>
<td></td>
<td>$\theta$-coordinate of intersection of streamline with blade leading edge, rad</td>
</tr>
<tr>
<td>THPC(11,NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>$\theta$-coordinates of points on alternate mesh (fig. 26), rad</td>
</tr>
<tr>
<td>THSL</td>
<td>TSONIN</td>
<td></td>
<td>$\theta$-coordinate (relative to MERIDL origin, not TSONIC origin) of mean blade surface at points along meridional streamlines, rad</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>THSP1(MM)</td>
<td></td>
<td>TSONIN</td>
<td>$\theta$-coordinates of blade suction-surface spline points, given as input for TSONIC (ref. 5), rad</td>
</tr>
<tr>
<td>THSP2(MM)</td>
<td></td>
<td>TSONIN</td>
<td>$\theta$-coordinates of blade pressure-surface spline points given as input to TSONIC, rad</td>
</tr>
<tr>
<td>THETOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>$\theta$-coordinates of intersections of horizontal mesh lines with blade trailing edge, rad</td>
</tr>
<tr>
<td>THTESL</td>
<td>TSONIN</td>
<td></td>
<td>$\theta$-coordinate of intersection of meridional streamline with blade trailing edge, rad</td>
</tr>
<tr>
<td>TINP</td>
<td>PRECAL</td>
<td>LOSSOM</td>
<td>$T_i$ $K$</td>
</tr>
<tr>
<td>TIP(NIN)</td>
<td>INPUTT</td>
<td></td>
<td>input $T_i$ at points along the line from hub to shroud on which upstream flow conditions are given, K</td>
</tr>
<tr>
<td>TIPBDY(MHTP1)</td>
<td>VBDRY</td>
<td></td>
<td>$T_i$ at points on upstream or downstream boundary of orthogonal mesh, K</td>
</tr>
<tr>
<td>TIPT(MHTP1)</td>
<td>TVELCY</td>
<td></td>
<td>$T_i$ at points along vertical mesh lines, K</td>
</tr>
<tr>
<td>TIPTEM</td>
<td>TSONIN</td>
<td></td>
<td>$T_i$ along a streamline, K, input TIP for TSONIC (ref. 5)</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>TITLEI(20)</td>
<td>INTITL</td>
<td>INPLOT</td>
<td>alphanumerical contents of input title card</td>
</tr>
<tr>
<td>TITL1(9)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INLET ABSOLUTE TOTAL TEMPERATURE</td>
</tr>
<tr>
<td>TITL1(15)</td>
<td></td>
<td>MEPL0T</td>
<td>plot title HUB, SHROUD, AND BLADE BOUNDARIES IN MERIDIONAL PLANE</td>
</tr>
<tr>
<td>TITL1(10)</td>
<td></td>
<td>SLPL0T</td>
<td>plot title STREAMLINE PLOT IN MERIDIONAL PLANE</td>
</tr>
<tr>
<td>TITL1(12)</td>
<td></td>
<td>SVPL0T</td>
<td>plot title MERIDIONAL AND SURFACE RELATIVE VELOCITIES</td>
</tr>
<tr>
<td>TITL2(8)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INLET ABSOLUTE TOTAL PRESSURE</td>
</tr>
<tr>
<td>TITL2(10)</td>
<td></td>
<td>MEPL0T</td>
<td>plot title ORTHOGONAL MESH IN MERIDIONAL PLANE</td>
</tr>
<tr>
<td>TITL2(3)</td>
<td></td>
<td>SLPL0T</td>
<td>plot title Z DIRECTION</td>
</tr>
<tr>
<td>TITL2(9)</td>
<td></td>
<td>SVPL0T</td>
<td>plot title STREAMLINE NO. XXXX, ( U = XXXXXXXX )</td>
</tr>
<tr>
<td>TITL3(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INLET ABSOLUTE WHIRL</td>
</tr>
<tr>
<td>TITL3(3)</td>
<td></td>
<td>MEPL0T</td>
<td>plot title Z DIRECTION</td>
</tr>
<tr>
<td>TITL3(3)</td>
<td></td>
<td>SLPL0T</td>
<td>plot title R DIRECTION</td>
</tr>
<tr>
<td>TITL3(14)</td>
<td></td>
<td>SVPL0T</td>
<td>plot title MERIDIONAL RELATIVE VELOCITIES FOR ALL STREAMLINES</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>TITL4(9)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INLET ABSOLUTE TANGENTIAL VELOCITY</td>
</tr>
<tr>
<td>TITL4(3)</td>
<td></td>
<td>MEPLOT</td>
<td>plot title R DIRECTION</td>
</tr>
<tr>
<td>TITL4(11)</td>
<td></td>
<td>SLPLOT</td>
<td>plot title SUBSONIC SOLUTION, ITERATION NO. XXXX</td>
</tr>
<tr>
<td>TITL4(15)</td>
<td></td>
<td>SVPLOT</td>
<td>plot title SUCTION SURFACE RELATIVE VELOCITIES FOR ALL STREAMLINES</td>
</tr>
<tr>
<td>TITL5(8)</td>
<td></td>
<td>INPLOT</td>
<td>plot title OUTLET ABSOLUTE TOTAL PRESSURE</td>
</tr>
<tr>
<td>TITL5(5)</td>
<td></td>
<td>SLPLOT</td>
<td>plot title TRANSONIC SOLUTION</td>
</tr>
<tr>
<td>TITL5(16)</td>
<td></td>
<td>SVPLOT</td>
<td>plot title PRESSURE SURFACE RELATIVE VELOCITIES FOR ALL STREAMLINES</td>
</tr>
<tr>
<td>TITL6(9)</td>
<td></td>
<td>INPLOT</td>
<td>plot title OUTLET ABSOLUTE TOTAL PRESSURE LOSS</td>
</tr>
<tr>
<td>TITL6(6)</td>
<td></td>
<td>SVPLOT</td>
<td>plot title MERIDIONAL COORDINATE</td>
</tr>
<tr>
<td>TITL7(6)</td>
<td></td>
<td>INPLOT</td>
<td>plot title OUTLET ABSOLUTE WHIRL</td>
</tr>
<tr>
<td>TITL7(2)</td>
<td></td>
<td>SVPLOT</td>
<td>plot title VELOCITY</td>
</tr>
<tr>
<td>TITL8(9)</td>
<td></td>
<td>INPLOT</td>
<td>plot title OUTLET ABSOLUTE TANGENTIAL VELOCITY</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>TITL10(13)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT BLADE SECTIONS FROM ZBL, RBL, THBL, TNBL</td>
</tr>
<tr>
<td>TITL11(6)</td>
<td></td>
<td>INPLOT</td>
<td>plot title BLADE SECTION NO. XXXX</td>
</tr>
<tr>
<td>TITL12(6)</td>
<td></td>
<td>INPLOT</td>
<td>plot title COMBINED BLADE SECTIONS</td>
</tr>
<tr>
<td>TITL13(4)</td>
<td></td>
<td>INPLOT</td>
<td>plot title STREAM FUNCTION</td>
</tr>
<tr>
<td>TITL14(2)</td>
<td></td>
<td>INPLOT</td>
<td>plot title RADIUS</td>
</tr>
<tr>
<td>TITL15(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - TIP</td>
</tr>
<tr>
<td>TITL16(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - PRIP</td>
</tr>
<tr>
<td>TITL17(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - LAMIN</td>
</tr>
<tr>
<td>TITL18(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - VTHIN</td>
</tr>
<tr>
<td>TITL19(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - PROP</td>
</tr>
<tr>
<td>TITL20(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - LOSOUT</td>
</tr>
<tr>
<td>TITL21(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - LAMOUT</td>
</tr>
<tr>
<td>TITL22(5)</td>
<td></td>
<td>INPLOT</td>
<td>plot title INPUT ARRAY - VTHOUT</td>
</tr>
<tr>
<td>TITL24(10)</td>
<td></td>
<td>INPLOT</td>
<td>plot title BLADE SECTION MERIDIONAL COORDINATE</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>TITL25(9)</td>
<td></td>
<td>INPLOT</td>
<td>plot title TANGENTIAL COORDINATE - RADIUS*THETA</td>
</tr>
<tr>
<td>TLEREF</td>
<td></td>
<td>TSONIN</td>
<td>6-coordinate of leading edge of blade, relative to TSONIC origin (ref. 5), rad</td>
</tr>
<tr>
<td>TNBL(NPPP,NBLPL)</td>
<td>INPUTT</td>
<td></td>
<td>input array of blade normal thicknesses, corresponding to ZBL,RBL coordinates, m</td>
</tr>
<tr>
<td>TOLER</td>
<td></td>
<td>MESHO</td>
<td>tolerance used in ROOT calls</td>
</tr>
<tr>
<td>TOLER</td>
<td></td>
<td>TIPF</td>
<td>tolerance for a point close to a spline point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RHOIPF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RHOOPF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LAMDAF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RVTHTA</td>
<td></td>
</tr>
<tr>
<td>TOM(MM, MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>t-coordinates of orthogonal mesh, m</td>
</tr>
<tr>
<td>TOP(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>T' o at points along vertical mesh lines, K</td>
</tr>
<tr>
<td>TOP</td>
<td></td>
<td>PRECAL</td>
<td>T' o, K</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOSSOM</td>
<td></td>
</tr>
<tr>
<td>TPP</td>
<td></td>
<td>NEWRHO</td>
<td>T'' at a mesh point, K</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TVELCY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OUTPUT</td>
<td></td>
</tr>
<tr>
<td>TPPN</td>
<td></td>
<td>TVELCY</td>
<td>new T'', K</td>
</tr>
<tr>
<td>TPPTIP(MM or MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>T''/T' i along vertical or horizontal mesh lines</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>TRNG</td>
<td>INPLOT</td>
<td></td>
<td>top or upper range of values on a given plot</td>
</tr>
<tr>
<td></td>
<td>SVPLOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTEM(NBLPL)</td>
<td>PRECAL</td>
<td></td>
<td>temporary storage of θ-coordinates, rad</td>
</tr>
<tr>
<td></td>
<td>THETOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTEREF</td>
<td>TSONIN</td>
<td></td>
<td>θ-coordinate of trailing edge of blade, relative to TSONIC origin (ref. 5), rad</td>
</tr>
<tr>
<td>TTIP</td>
<td>VBDRY</td>
<td>T/T_i</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NEWRHO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLDVEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TVELCY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TVERT(MHTP1)</td>
<td>NEWRHO</td>
<td>temporary storage for values from TOM array on a vertical mesh line, m</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLDVEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TVELCY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWLMR(MHTP1)</td>
<td>TVELCY</td>
<td>2ωλ - (wr)^2 at points along vertical mesh lines, m^2/sec^2</td>
<td></td>
</tr>
<tr>
<td>TWLMR</td>
<td>VBDRY</td>
<td>2ωλ - (wr)^2, m^2/sec^2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NEWRHO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLDVEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UBDEV</td>
<td>INDEV</td>
<td>deviation angle, neglecting blockage correction, where horizontal orthogonal intersects blade, (β_fs - β_b)_le, deg</td>
<td></td>
</tr>
<tr>
<td>UBINC</td>
<td>INDEV</td>
<td>incidence angle, neglecting blockage correction, where horizontal orthogonal intersects blade, (β_fs - β_b)_le, deg</td>
<td></td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>UILOM(MHTP1)</td>
<td></td>
<td>LAMDAF</td>
<td>stream-function values for spline fit of stream function against radius</td>
</tr>
<tr>
<td>UNEW</td>
<td></td>
<td>SOR</td>
<td>new estimate for ( u ) at a mesh point</td>
</tr>
<tr>
<td>UOLOM(MHTP1)</td>
<td></td>
<td>RVTHTA</td>
<td>stream-function values for spline fit of stream function against radius</td>
</tr>
<tr>
<td>UOM(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>stream function, ( u ), at orthogonal mesh points</td>
</tr>
<tr>
<td>UTEM(MHTP1 or 20)</td>
<td></td>
<td>OUTPUT</td>
<td>temporary storage for values from UOM array on vertical mesh lines; also stream function at 20 equally spaced points from hub to shroud</td>
</tr>
<tr>
<td>UVERT(MHTP1, 2)</td>
<td></td>
<td>SOR</td>
<td>temporary storage for boundary values of stream function on upstream and downstream boundaries of orthogonal mesh</td>
</tr>
<tr>
<td>UVERT(MHTP1)</td>
<td></td>
<td>NEWRHO</td>
<td>temporary storage for values from UOM array along vertical mesh lines</td>
</tr>
<tr>
<td>VELTOL</td>
<td></td>
<td>INPUTT</td>
<td>input convergence tolerance on maximum velocity change in each outer iteration, over all mesh points, for reduced mass flow</td>
</tr>
<tr>
<td>VTH(MM, MHTP1)</td>
<td></td>
<td>VARCOM</td>
<td>( V_\theta ) at orthogonal mesh points, m/sec</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>VTHIN(NIN)</td>
<td>INPUTT</td>
<td></td>
<td>input values of $(V_\theta)^i$ at points along line from hub to shroud on which upstream flow conditions are given, m/sec</td>
</tr>
<tr>
<td>VTHOUT(NOUT)</td>
<td>INPUTT</td>
<td></td>
<td>input values of $(V_\theta)^0$ at points along line from hub to shroud on which downstream flow conditions are given, m/sec</td>
</tr>
<tr>
<td>W(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>W at orthogonal mesh points, m/sec</td>
</tr>
<tr>
<td>WAS</td>
<td>VBDRY</td>
<td>TVELCY</td>
<td>first estimate of $W_{j+1}$ at next mesh point along vertical mesh line (eq. (1)), $W^*_{j+1}$, m/sec</td>
</tr>
<tr>
<td>WASS</td>
<td>VBDRY</td>
<td>TVELCY</td>
<td>second estimate of $W_{j+1}$ at next mesh point along vertical mesh line (eq. (1)), $W^{**}_{j+1}$, m/sec</td>
</tr>
<tr>
<td>WBDRY(MHTP1)</td>
<td>VBDRY</td>
<td></td>
<td>W on upstream or downstream boundary of orthogonal mesh, calculated by velocity-gradient equation ([C9], part I), m/sec</td>
</tr>
<tr>
<td>WFLF</td>
<td>VBDRY</td>
<td></td>
<td>$\left(\frac{r^2 - r_{hub}^2}{r_{tip}^2 - r_{hub}^2}\right)$ area fraction, estimate of stream function at radius, r</td>
</tr>
<tr>
<td>WHIRL</td>
<td>TVELCY</td>
<td></td>
<td>$\lambda$ or $(rV_\theta)^o$, m$^2$/sec</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>WHUB</td>
<td></td>
<td>VBDRY</td>
<td>estimate of $W_{hub}$, m/sec</td>
</tr>
<tr>
<td>WLSSL(MM, NSL)</td>
<td>SLCOM</td>
<td>TVELCY</td>
<td>$W_L$ at points along streamlines where they cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WLSST(NSL, NOSTAT)</td>
<td>SLCOM</td>
<td>STACOM</td>
<td>$W_L$ at points along station lines where they cross streamlines, m/sec</td>
</tr>
<tr>
<td>WLSURF(MM, MHTP1)</td>
<td>VARCOM</td>
<td>SLCOM</td>
<td>$W_L$ on orthogonal mesh, m/sec</td>
</tr>
<tr>
<td>WMSL(MM, NSL)</td>
<td>SLCOM</td>
<td>STACOM</td>
<td>$W_m$ at points where streamlines cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WMST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td>SLCOM</td>
<td>$W_m$ at points where station lines cross streamlines, m/sec</td>
</tr>
<tr>
<td>WMVERT(MHTP1)</td>
<td></td>
<td>TVELCY</td>
<td>temporary storage for values from WSUBM array on vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WRSI(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$W_r$ at points where streamlines cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WRST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td>SLCOM</td>
<td>$W_r$ at points where station lines cross streamlines, m/sec</td>
</tr>
<tr>
<td>WSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$W$ at points where streamlines cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WSM</td>
<td></td>
<td>VBDRY</td>
<td>$W_s$ at mean radius, m/sec</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>WSQ</td>
<td></td>
<td>VBDRY NEWRHO</td>
<td>( W^2, \text{m}^2/\text{sec}^2 )</td>
</tr>
<tr>
<td>WSQ</td>
<td></td>
<td>TVELCY</td>
<td></td>
</tr>
<tr>
<td>WST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>( W_\text{at points where station lines cross streamlines, } \text{m/sec} )</td>
</tr>
<tr>
<td>WSUBM(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_m \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WSUBR(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_r \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WSUBS(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_s \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WSUBT(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_t \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WSUBZ(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_z \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WTEMP</td>
<td></td>
<td>NEWRHO</td>
<td>new calculated value of ( W ) at a mesh point, ( \text{m/sec} )</td>
</tr>
<tr>
<td>WTFL</td>
<td></td>
<td>TSONIN</td>
<td>mass flow per blade in a stream sheet, ( \text{kg/sec} ), input to TSONIC (ref. 5)</td>
</tr>
<tr>
<td>WTH(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>( W_\theta \text{ at orthogonal mesh points, } \text{m/sec} )</td>
</tr>
<tr>
<td>WTHETA</td>
<td></td>
<td>VBDRY TVELCY</td>
<td>( W_\theta, \text{m/sec} )</td>
</tr>
<tr>
<td>WTHSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>( W_\theta \text{ at points where streamlines cross vertical mesh lines, } \text{m/sec} )</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WTHST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$W_\theta$ at points where station lines cross streamlines, m/sec</td>
</tr>
<tr>
<td>WTSSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$W_{tr}$ at points where streamlines cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WTSST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$W_{tr}$ at points where station lines cross streamlines, m/sec</td>
</tr>
<tr>
<td>WTSURF(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$W_{tr}$ on orthogonal mesh, m/sec</td>
</tr>
<tr>
<td>WTVERT(MHTP1)</td>
<td>TVELCY</td>
<td></td>
<td>WTH on vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WWCR(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$W/W_{cr}$ at orthogonal mesh points</td>
</tr>
<tr>
<td>WWCRSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$W/W_{cr}$ at points where streamlines cross vertical mesh lines</td>
</tr>
<tr>
<td>WWCRST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$W/W_{cr}$ at points where station lines cross streamlines</td>
</tr>
<tr>
<td>WZSL(MM, NSL)</td>
<td>SLCOM</td>
<td></td>
<td>$W_z$ at points where streamlines cross vertical mesh lines, m/sec</td>
</tr>
<tr>
<td>WZST(NSL, NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>$W_z$ at points where station lines cross streamlines, m/sec</td>
</tr>
<tr>
<td>XIOM(MM, MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$\xi$ at orthogonal mesh points, (eq. (A2)), 1/m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>XIOMT</td>
<td></td>
<td>NEWRHO</td>
<td>new estimated value of $\xi$ at a mesh point</td>
</tr>
<tr>
<td>XNEW</td>
<td></td>
<td>NEWRHO</td>
<td>percentage of new calculated value of XIOMT used in updating XIOM</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td>CROSCD</td>
<td>reference z-coordinate, m</td>
</tr>
<tr>
<td>Z1</td>
<td></td>
<td>MESHO</td>
<td>z-coordinate of intersection of line 1, fig. 8, with upper horizontal mesh line, m</td>
</tr>
<tr>
<td>Z2</td>
<td></td>
<td>MESHO</td>
<td>z-coordinate of intersection of line 2, fig. 8, with upper horizontal mesh line, m</td>
</tr>
<tr>
<td>ZBL(NPPP,NBLPL)</td>
<td>INPUTT</td>
<td></td>
<td>input array of z-coordinates of points describing blade surface, m</td>
</tr>
<tr>
<td>ZBLTEM(NBLPL)</td>
<td>OUTPUT</td>
<td></td>
<td>temporary storage for values in ZLE and ZTE arrays, m</td>
</tr>
<tr>
<td>ZEROM</td>
<td>OUTPUT</td>
<td></td>
<td>translation distance on m-coordinate so that $m = 0$ corresponds to $z = 0$, m</td>
</tr>
<tr>
<td>ZETOM(MM,MHTP1)</td>
<td>VARCOM</td>
<td></td>
<td>$\xi$ at orthogonal mesh points (eq. (A3)), m/sec$^2$</td>
</tr>
<tr>
<td>ZETOMT</td>
<td>NEWRHO</td>
<td></td>
<td>new estimated value of $\xi$ at a mesh point</td>
</tr>
</tbody>
</table>

112
<table>
<thead>
<tr>
<th>Variable name</th>
<th>COMMON block</th>
<th>Subroutine</th>
<th>Description and comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZHIN</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection with hub profile of line on which upstream flow conditions are given, m</td>
</tr>
<tr>
<td>ZHOUT</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection with hub profile of line on which downstream flow conditions are given, m</td>
</tr>
<tr>
<td>ZHPLT(100)</td>
<td>PLTCOM</td>
<td></td>
<td>z-coordinates used for plotting hub profile, ZHUB, in MEPLOT, m</td>
</tr>
<tr>
<td>ZHST(NOSTAT)</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinates of intersections of hub-shroud output station lines with hub profile, m</td>
</tr>
<tr>
<td>ZHUB(NHUB)</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinates of points defining hub or bottom boundary of flow channel, m</td>
</tr>
<tr>
<td>ZILOM</td>
<td>LAMDAF</td>
<td></td>
<td>z-coordinate corresponding to RILOM</td>
</tr>
<tr>
<td>ZLE(NBLPL)</td>
<td>CALCON</td>
<td></td>
<td>z-coordinates of input blade section points (from ZBL) defining leading edge of blade, m</td>
</tr>
<tr>
<td>ZLEH</td>
<td>PRECAL</td>
<td></td>
<td>z-coordinate of intersection of leading edge of blade with hub profile, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>
| ZLEOM(MHTP1)      | CALCON       |            | z-coordinates of intersec-
|                   |              |            | tions of horizontal mesh |
|                   |              |            | lines with blade leading |
|                   |              |            | edge, m                  |
| ZLESL             | TSONIN       |            | z-coordinate of intersec-
|                   |              |            | tion of a streamline with |
|                   |              |            | leading edge of blade, m  |
| ZLET              | PRECAL       |            | z-coordinate of intersec-
|                   |              |            | tion of leading edge of blad |
|                   |              |            | e with shroud profile, m  |
| ZLPLT(100)        | PLTCOM       |            | z-coordinates used for plot-
|                   |              |            | ting blade leading edge, ZLE, |
|                   |              |            | in MEPLOT, m              |
| ZLRNG             | PLTCOM       |            | z-coordinate of left-hand |
|                   |              |            | boundary of a plot of meri-
|                   |              |            | dional plane or orthogonal |
|                   |              |            | mesh, done in MEPLOT, m   |
| ZNEW              | NEWRHO       |            | percentage of new calculated |
|                   |              |            | value of ZETOMT used in   |
|                   |              |            | updating ZETOM             |
| ZOLOM             | RVHTHA       |            | z-coordinate corresponding|
|                   |              |            | to ROLOM                  |
| ZOM(MM, MHTP1)    | CALCON       |            | z-coordinates of orthogonal |
|                   |              |            | mesh, m                   |
| ZOMBI             | INPUTT       |            | input z-coordinate of inter-
<p>|                   |              |            | section of vertical mesh   |
|                   |              |            | line with hub profile where |
|                   |              |            | first change in mesh       |
|                   |              |            | spacing occurs (MBI), m    |</p>
<table>
<thead>
<tr>
<th>Variable name</th>
<th>COMMON block</th>
<th>Subroutine</th>
<th>Description and comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOMBO</td>
<td>INPUTT</td>
<td>Block</td>
<td>input z-coordinate of intersection of vertical mesh line with hub profile where second change in mesh spacing occurs (MBO), m</td>
</tr>
<tr>
<td>ZOMIN</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection of left boundary of orthogonal mesh with hub profile, m</td>
</tr>
<tr>
<td>ZOMOUT</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection of right boundary of orthogonal mesh (MM) with hub profile, m</td>
</tr>
<tr>
<td>ZPC(11,NBLPL)</td>
<td>INDCOM</td>
<td></td>
<td>z-coordinates of points of alternate mesh (fig. 26)</td>
</tr>
<tr>
<td>ZRRNG</td>
<td>PLTCOM</td>
<td></td>
<td>z-coordinate of right-hand boundary of a plot of meridional plane or orthogonal mesh, done in MEPLT, m</td>
</tr>
<tr>
<td>ZSL(MM,NSL)</td>
<td>SLCOM</td>
<td></td>
<td>z-coordinates of points where streamlines cross vertical mesh lines, m</td>
</tr>
<tr>
<td>ZSLTEM(NSL)</td>
<td>OUTPUT</td>
<td></td>
<td>temporary storage for calculated values to be put into ZSL array, m</td>
</tr>
<tr>
<td>ZSPL</td>
<td>ILETE</td>
<td></td>
<td>z-coordinate on leading or trailing edge of blade corresponding to a streamline, m</td>
</tr>
<tr>
<td>ZSPLT(100)</td>
<td>PLTCOM</td>
<td></td>
<td>z-coordinates used for plotting shroud profile, ZTIP, in MEPLT, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>ZST(NSL,NOSTAT)</td>
<td>STACOM</td>
<td></td>
<td>z-coordinates of points where station lines cross streamlines, m</td>
</tr>
<tr>
<td>ZTE(NBLPL)</td>
<td>CALCON</td>
<td></td>
<td>z-coordinates of input blade section points (from ZBL) defining trailing edge of blade, m</td>
</tr>
<tr>
<td>ZTEH</td>
<td>PRECAL</td>
<td></td>
<td>z-coordinate of intersection of trailing edge of blade with hub profile, m</td>
</tr>
<tr>
<td>ZTEM(10)</td>
<td>INPUT</td>
<td></td>
<td>temporary storage for z-coordinate, m</td>
</tr>
<tr>
<td>ZTEM(MM)</td>
<td>MEPLIT</td>
<td></td>
<td>temporary storage of z-coordinates from ZOM for plotting, m</td>
</tr>
<tr>
<td>ZTEM(NBLPL)</td>
<td>THETOM</td>
<td></td>
<td>temporary storage of z-coordinates from ZPC, m</td>
</tr>
<tr>
<td>ZTEM(MHTP1 or NOSTAT or 20)</td>
<td>OUTPUT</td>
<td></td>
<td>temporary storage for values from ZOM array on vertical mesh lines; also temporary storage for values from ZST array along station lines, m</td>
</tr>
<tr>
<td>ZTEOM(MHTP1)</td>
<td>CALCON</td>
<td></td>
<td>z-coordinates of intersections of horizontal mesh lines with blade trailing edge, m</td>
</tr>
<tr>
<td>ZTESL</td>
<td>TSONIN</td>
<td></td>
<td>z-coordinate of intersection of a streamline with trailing edge of blade, m</td>
</tr>
<tr>
<td>Variable name</td>
<td>COMMON block</td>
<td>Subroutine</td>
<td>Description and comments</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>ZTET</td>
<td>PRECAL</td>
<td></td>
<td>z-coordinate of intersection of trailing edge of blade with shroud profile, m</td>
</tr>
<tr>
<td>ZTIN</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection of line on which upstream flow conditions are given with shroud profile, m</td>
</tr>
<tr>
<td>ZTIP(NTIP)</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinates of points defining shroud or top boundary of flow channel, m</td>
</tr>
<tr>
<td>ZTOUT</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinate of intersection of line on which downstream flow conditions are given with shroud profile, m</td>
</tr>
<tr>
<td>ZTPLT(100)</td>
<td>PLTCOM</td>
<td></td>
<td>z-coordinate used for plotting blade trailing edge, ZTE, in MEPLOT, m</td>
</tr>
<tr>
<td>ZTST(NOSTAT)</td>
<td>INPUTT</td>
<td></td>
<td>input z-coordinates of intersections of hub-shroud output station lines with shroud profile, m</td>
</tr>
</tbody>
</table>
PROGRAM LISTING

COMMON SRW, SRE, ITER, IEND, NREAD, NWRIT
COMMON INPUT, CAM, AR, MSPL, OMEGA, REDFAC, VELTOL, FNEW, DNEW, MBI, MBO,
1 W, MHT, NBL, NHUR, NTIP, NI, NCUT, NLPL, NAPP, NOST, AT, NSL, LSFR,
2 LPL, LAMVT, IMESH, ISLINE, ISTATL, IPLOT, ISUPER, ITSON, IDBUG,
3 ZMIN, ZMAX, ZCPC, ZCOUT, ZIN, ZT IN, ZHOUT, ZOUT, ZHT(50),
4 RHUR(50), TIP(50), RTIP(50), SFIN(50), RADIUS(50), TIP(50), PRIP(50),
5 LAMIN(50), VTHIN(50), SFOUT(50), RADOUT(50), PRIP(50), LOSOUT(50),
6 LAMOUT(50), VTHOUT(50), ZHST(50), ZTST(50), FLFR(50),
7 ZR(50, 50), RR(50, 50), TB(50, 50), TNL(50, 50)
EXTERNAL TIP, TOPF, RHOIFP, RHOOPF, LAMDAF, RVTHTA
INTEGER SRW, SRE
10 IEND=-1
ITER=0
C
C--READ AND PLOT INPUT DATA
CALL INPUT
CALL INPLOT
C
C--GENERATE ORTHOGONAL MESH
CALL MESHO
C
C--CALCULATE ALL PRELIMINARY FIXED CONSTANTS
CALL PRECAL
C
C--PLOT ORTHOGONAL MESH
CALL MEPLT
C
C--CALCULATE STREAM FUNCTION ON UPSTREAM AND DOWNSTREAM
C--BOUNDARIES OF THE ORTHOGONAL MESH
CALL VRDRY(1, TIP, RHOIFP, LAMDAF)
CALL VRDRY (MM, TOPF, RHOOPF, RVTHTA)
C
C--CALCULATE COEFFICIENTS, SOLVE DIFFERENTIAL EQUATIONS FOR STREAM
C--FUNCTION, AND COMPUTE NEW VELOCITIES AND DENSITIES
CALL INIT
20 ITER = ITER+1
CALL CCEF
CALL SCR
CALL NEWRHO
C
C--CALCULATE AND PRINT MAJOR OUTPUT DATA
CALL OUTPUT
CALL INDEV
C
C--PLOT STREAMLINES AND PLOT VELOCITIES
CALL SLPLOT
CALL SVPLT
IF (IEND.LT.0) GO TO 20
IF (REDFAC.EQ.1.0) GO TO 10
C
C--OBTAIN TRANSONIC SOLUTION WITH FULL MASS FLOW
30 CALL TVELC
IF (ISUPER.EQ.2) GO TO 10
REDFAC = 1.0

118
CALL TCTPT
CALL TNCST
CALL SLPLOT
CALL SVPLLOT
IF (ISUPER.EQ.0) GC TO 10
ISUPER = 0
GC TC 30
END

SUBRoutines INPUT

C--INPUT READS AND PRINTS ALL INPUT DATA CARDS
C
C   COMMON SRW, SRE, ITER, IENC, NREAD, NWRITE
C   COMMON/INPUT/GAM, AR, MSFL, CMGEA, REDFAC, VELTOL, FNEW, DNEW, MBI, MBO,
1   MM, MHT, NBL, HUB, NTIP, NIN, NCUT, NLRPL, NPPP, NSTAT, NSL, LSFR,
2   LTPL, LAMVT, IMESH, ISLNE, ISTATL, PLOT, ISUPER, ITSON, IDEBUG,
3   TCMIN, TCMAX, TCMON, TCMON, TFMN, TFIT, TFIN, THOUT, TCTOUT, TCTOUT,
4   RHUR(50), RTIP1(50), SFIN(50), RADIN(50), TIP(50), PRIP(50),
5   LAMIN(50), VTHIN(50), SFOUT(50), RADOUT(50), PROP(50), LSOUT(50),
6   LAMOUT(50), VTHOUT(50), ZHTST(50), ZTST(50), FLFR(50),
7   ZPL(50,50), RBL(50,50), THBL(50,50), THNL(50,50),

   COMMON/CALCCN/MM1, MHTPL1, CP, EXPOX, TPROG, PITCH, CURVHI, CURVTI,
   CURVHO, CURVTO, RHIN, RHIN, RHOUT, RLEH, RLET, RTET,
1   RLE(50), RLET(50), RLET(50), RTE(50), RLET(101), RLET(101),
2   SLEM(101), SLHECM(101), SLEM(101), RLECM(101),
3   TTM(101), ITLE(101), ITLE(101), ITLE(101),
4   TTM(100,101), ITLE(100,101), ITLE(100,101), ITLE(100,101),
5   TTM(100,101), ITLE(100,101), ITLE(100,101), ITLE(100,101),
6   TTM(100,101), ITLE(100,101), ITLE(100,101), ITLE(100,101),

   COMMON/INTITL/TITLE(20),
   DIMENSION 7TEM(10), RTEM(10), DYDX(10),
   REAL MSFL, LAMIN, LAMOUT, LCSOUT

C--READ AND PRINT INPUT DATA
C
NREAD = 5
NWRITE = 6
10 WRITE(NWRITE, 1000) (TITLE(I), I=1,20)
WRITE(NWRITE, 1050) (TITLE(I), I=1,20)
WRITE(NWRITE, 1100)
READ (NREAD, 1030) GAM, AR, MSFL, CMGEA, REDFAC, VELTOL, FNEW, DNEW,
IF (REDFAC.LE.0.) REDFAC=1.0
IF (VELTOL.LE.0.) VELTOL=0.01
IF (FNEW.LE.0.) FNEW=0.5
IF (DNEW.LE.0.) DNEW=0.5
VELTOL = VELTOL*AMTN1(FNEW, DNEW)
WRITE(NWRITE, 1040) GAM, AR, MSFL, CMGEA, REDFAC, VELTOL, FNEW, DNEW
WRITE(NWRITE, 1110)
READ (NREAD, 1010) MBI, MBI, MBI, MM, MHT, NBL, HUB, NTIP, NIN, NCUT, NLRPL,
1   NPPP, NSTAT, NSL
WRITE(NWRITE, 1020) MBI, MBI, MBI, MM, MHT, NBL, HUB, NTIP, NIN, NCUT, NLRPL,
1   NPPP, NSTAT, NSL
WRITE(NWRITE, 1120)
READ (NREAD, 1010) LSFR, LTPL, LAMVT

119
WRITE(NWRITE,1020) LSFR,LTPL,LAMVT
WRITE(NWRITE,1130)
READ (NREAD,1030) ZOMIN,ZOMAI,ZOMAO,ZOMOUT
WRITE(NWRITE,1040) ZOMIN,ZOMAI,ZOMAO,ZOMOUT
WRITE(NWRITE,1140)
READ (NREAD,1030) (ZHUR(I),I=1,NHUB)
WRITE(NWRITE,1040) (ZHUR(I),I=1,NHUB)
WRITE(NWRITE,1150)
READ (NREAD,1030) (RHUB(I),I=1,NHUB)
WRITE(NWRITE,1040) (RHUB(I),I=1,NHUB)
WRITE(NWRITE,1160)
READ (NREAD,1030) (TIP(I),I=1,NTIP)
WRITE(NWRITE,1040) (TIP(I),I=1,NTIP)
WRITE(NWRITE,1170)
READ (NREAD,1030) (RTIP(I),I=1,NTIP)
WRITE(NWRITE,1040) (RTIP(I),I=1,NTIP)
WRITE(NWRITE,1180)
READ (NREAD,1030) ZHIN,ZTIN
WRITE(NWRITE,1040) ZHIN,ZTIN
IF (LSFR.EQ.1) GO TO 20
WRITE(NWRITE,1190)
READ (NREAD,1030) (SFIN(I),I=1,NIN)
WRITE(NWRITE,1040) (SFIN(I),I=1,NIN)
GO TO 30
20 WRITE(NWRITE,1200)
READ (NREAD,1030) (RADIN(I),I=1,NIN)
WRITE(NWRITE,1040) (RADIN(I),I=1,NIN)
WRITE(NWRITE,1210)
READ (NREAD,1030) (TIP(I),I=1,NIN)
WRITE(NWRITE,1040) (TIP(I),I=1,NIN)
WRITE(NWRITE,1220)
READ (NREAD,1030) (PRIP(I),I=1,NIN)
WRITE(NWRITE,1040) (PRIP(I),I=1,NIN)
IF (LAMVT.EQ.1) GO TO 40
WRITE(NWRITE,1230)
READ (NREAD,1030) (LAMIN(I),I=1,NIN)
WRITE(NWRITE,1040) (LAMIN(I),I=1,NIN)
GO TO 50
40 WRITE(NWRITE,1240)
READ (NREAD,1030) (VTHIN(I),I=1,NIN)
WRITE(NWRITE,1040) (VTHIN(I),I=1,NIN)
WRITE(NWRITE,1250)
READ (NREAD,1030) ZHOUT,ZTOUT
WRITE(NWRITE,1040) ZHOUT,ZTOUT
IF (LSFR.EQ.1) GO TO 60
WRITE(NWRITE,1260)
READ (NREAD,1030) (SFOUT(I),I=1,NOUT)
WRITE(NWRITE,1040) (SFOUT(I),I=1,NOUT)
GO TO 70
60 WRITE(NWRITE,1270)
READ (NREAD,1030) (RADOUT(I),I=1,NOUT)
WRITE(NWRITE,1040) (RADOUT(I),I=1,NOUT)
70 IF (LTPL.EQ.1) GO TO 80
WRITE(NWRITE,1280)
READ (NREAD,1030) (PROP(I),I=1,NOUT)
WRITE(NWRITE,1040) (PROP(I),I=1,NOUT)
GO TO 90
80 WRITE(NWRITE,1290)
READ (NREAD,1030) (LOSOUT(I),I=1,NOUT)
WRITE(NWRITE,1040) (LOSOUT(I),I=1,NOUT)
GO TO 70
50 IF (LAMVT, F.C. 1) GO TO 100
WRITE(NWRIT, 1300)
READ (NREAD, 1030) (LAMOUT(I), I=1, NOUT)
WRITE(NWRIT, 1040) (LAMOUT(I), I=1, NOUT)
GO TO 110
100 WRITE(NWRIT, 1310)
READ (NREAD, 1030) (VTHOUT(I), I=1, NOUT)
WRITE(NWRIT, 1040) (VTHOUT(I), I=1, NOUT)
110 WRITE(NWRIT, 1320)
DO 120 JN=1, NRPPL
READ (NREAD, 1030) (ZBL(IN, JN), IN=1, NPPP)
120 WRITE(NWRIT, 1330)
CC 130 JN=1, NRPPL
READ (NREAD, 1030) (RBL(IN, JN), IN=1, NPPP)
130 WRITE(NWRIT, 1340)
DC 140 JN=1, NRPPL
READ (NREAD, 1030) (THBL(IN, JN), IN=1, NPPP)
140 WRITE(NWRIT, 1350)
DO 150 JN=1, NRPPL
READ (NREAD, 1030) (TNBL(IN, JN), IN=1, NPPP)
150 WRITE(NWRIT, 1040)
IF (NOSTAT.EQ.0) GO TO 160
WRITE(NWRIT, 1360)
READ (NREAD, 1030) (ZHST(I), I=1, NCSTAT)
WRITE(NWRIT, 1040) (ZHST(I), I=1, NCSTAT)
WRITE(NWRIT, 1370)
READ (NREAD, 1030) (ZUST(I), I=1, NOSTAT)
WRITE(NWRIT, 1040) (ZUST(I), I=1, NOSTAT)
160 IF (NSL.EQ.0) GO TO 170
WRITE(NWRIT, 1380)
READ (NREAD, 1030) (FLFR(I), I=1, NSL)
WRITE(NWRIT, 1040) (FLFR(I), I=1, NSL)
170 WRITE(NWRIT, 1390)
READ (NREAD, 1010) IMESH, ISLINE, ISTATL, ILOT, ISUPER, ITSON, IDIBUG
WRITE(NWRIT, 1020) IMESH, ISLINE, ISTATL, ILOT, ISUPER, ITSON, IDIBUG
WRITE(NWRIT, 1000)
IF (MM.LE.100. AND. MHT.LE.100. AND. NHU8.LE.50. AND. NTP.LE.50. AND.
IN.IN.LE.50. AND. NOUT.LE.50. AND. NRPPL.LE.50. AND. NPPP.LE.50. AND.
2NCSTAT.LE.50. AND. NSL.LE.50. AND. LSFR.GE.0. AND. LSFR.LE.1. AND.
3LTP.GE.0. AND. LTPL.LE.1. AND. LAMVT.GE.0. AND. LAMVT.LE.1. GO TO 180
WRITE(NWRIT, 1400)
STOP
C
C--CALCULATE MISCELLANEOUS CONSTANTS
C
180 MMM1 = MM-1
MHTP1 = MHT+1
EXPCN = 1.0/(CAM-1.)
CP = AR*CAM*EXPCN
TCRCG = 2.*CAM*AR/(CAM+1.)
PITCH = 2.*3.1415927/FLOAT(N8L)
MSFL = MSFL/FLOAT(N8L)
C
C--ASSUME VALUES FOR ZHIN, ZTIN, ZHOUT, AND ZTOUT
C--IF THEY WERE NOT GIVEN AS INPUT
C
IF (LSFR.EQ.1) GO TO 200
IF (ZHIN.NE.0.0 OR ZTIN.NE.0.) GC TO 190
ZTIN = Z0MIN
ZHTIN = Z0MIN
160 IF (ZTOUT.NE.0.0 OR ZTOUT.NE.0.) GC TO 200
ZTOUT = Z0MOUT
ZTCUT = Z0MCUT
C
C--CALCULATE ESTIMATED UPSTREAM AND DOWNSTREAM VALUES OF C--STREAM FUNCTION, IF RADIUS WAS GIVEN AS INPUT
C
200 ZTEM(1) = ZHIN
ZTEM(2) = ZTCUT
CALL SPLINT(ZHUB, Rhub, NHUB, ZTEM, 2, RTEM, CYDX)
RHIN = RTEM(1)
RHOUT = RTEM(2)
ZTEM(1) = ZTIN
ZTEM(2) = ZTCUT
CALL SPLINT(ZTIP, RTIP, NTIP, ZTEM, 2, RTEM, DYDX)
RTIN = RTEM(1)
RTOUT = RTEM(2)
RINSC = RTIN**2 - RHIN**2
ROUTSQ = RTOUT**2 - RHOUT**2
IF (LSFR.EQ.0) GO TO 230
DO 210 J=1, NIN
210 SFIN(J) = (RADIN(J)**2 - RHIN**2) / RINSC
DO 220 J=1, NOUT
220 SFOUT(J) = (RADOUT(J)**2 - RHOUT**2) / ROUTSQ
GO TO 260
C
C--CALCULATE ESTIMATED UPSTREAM AND DOWNSTREAM VALUES OF C--RADIUS, IF STREAM FUNCTION WAS GIVEN AS INPUT
C
230 DC 240 J=1, NIN
240 RADIN(J) = SQRT(RHIN**2 + SFIN(J)*RINSC)
DO 250 J=1, NOUT
250 RADOUT(J) = SQRT(RHOUT**2 + SFOUT(J)*ROUTSQ)
C
C--CALCULATE ESTIMATED UPSTREAM AND DOWNSTREAM TANGENTIAL VELOCITIES, C--IF WHIRL WAS GIVEN AS INPUT
C
260 IF (LAMVT.EQ.1) GO TO 290
DO 270 J=1, NIN
270 VTHIN(J) = LAMIN(J) / RADIN(J)
DO 280 J=1, NOUT
280 VTHOUT(J) = LAMOUT(J) / RADOUT(J)
IF (LSFR.EQ.1) LAMVT=1
RETURN
C
C--CALCULATE ESTIMATED UPSTREAM AND DOWNSTREAM WHIRL, C--IF TANGENTIAL VELOCITY WAS GIVEN AS INPUT
C
290 DO 300 J=1, NIN
300 LAMIN(J) = RADIN(J) * VTHIN(J)
DO 310 J=1, NOUT
310 LAMOUT(J) = RADOUT(J) * VTHOUT(J)
C
C--FORMAT STATEMENTS
C
122
SUBROUTINE INPLCT

C--INPLOT PLOTS THE UPSTREAM AND DOWNSTREAM INPUT FLOW VARIABLES
C--AS WELL AS THE INPUT BLADE SECTIONS FROM HUB TO SPROUD
C
C-- PLOT TITLE ON MICROFILM

CALL LRCHS7(4)
CALL LRXLGN(TITLEI,80,0,1.0,5.0,1.0)
CALL LRCHS7(2)
CALL LRCHS7(1)
CALL LRLEGN(TITLEI,80,0,1.0,0.0,0.0)
CALL LRMOFF

C-- PREPARE FOR PLOTTING OF INLET CONDITIONS

IF (LSFR.EQ.1) GO TO 20
PLTY(I) = SFIN(I)
PLTY(101) = SFOUT(1)
DEL = (SFIN(NIN) - SFIN(1))/100.
DC 10 J=2,100
10 PLTY(J) = PLTY(J-1) + DEL
BRNG = AMIN1(SFIN(1),SFOUT(1))
TRNG = AMAX1(SFIN(NIN),SFOUT(NCUT))
GO TO 40
20 PLTY(I) = RADIN(I)
PLTY(101) = RADIN(NIN)
DEL = (RADIN(NIN) - RADIN(1))/100.
DC 30 J=2,100
30 PLTY(J) = PLTY(J-1) + DEL
BRNG = AMIN1(RADIN(I),RADOUT(I))
TRNG = AMAX1(RADIN(NIN),RADOUT(NOUT))

C-- PLOT INLET ABSOLUTE TOTAL TEMPERATURE

40 LRNG = TIP(1)
RRNG = TIP(1)
DC 50 J=1,NIN
LRNG = AMIN1(LRNG,TIP(J))
50 RRNG = AMAX1(RRNG,TIP(J))
CALL LRGRDN(1.0,1.0,2.0,1.0)
CALL LRANGE(LRNG,RRNG,BRNG,TRNG)
CALL LRGRID(1.1,11.0,11.0)
CALL LRCHS7(4)
CALL LRLEGN(TITL,36,0.1,0.0,5.0,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL15,20,0.4,0.1,3.0,0.0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,0.2,4.2,0.0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1.0,2.4,7.0,0.0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFIN,TIP,NIN,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RADIN,TIP,NIN,PLTY,101,PLTX,GRAD)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0.0)
IF (LSFR.EQ.0) CALL LRCURV(TIP,SFIN,NIN,4,SYM,1.0)
IF (LSFR.EQ.1) CALL LRCURV(TIP,RADIN,NIN,4,SYM,1.0)

C-- PLOT INLET ABSOLUTE TOTAL PRESSURE

LRNG = PRIP(1)
RRNG = PRIP(1)
DC 60 J=1,NIN

125
LRNG = AMIN(LRNG,PRIP(J))
60 RRNG = AMAX1(RRNG,PRIP(J))
CALL LRANGE(LRNG,RRNG,RRNG,TRNG)
CALL LRLEGN(TITL2,32,0,1.6,0.5,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL16,20,0,4.0,1.3,0.0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,1,0.2,4.2,0.0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1,0.2,4.7,0.0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFIN,PRIP,NIN,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RADIN,PRIP,NIN,PLTY,101,PLTX,GRAD)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0.0)
IF (LSFR.EQ.0) CALL LRCURV(PRIP,SFIN,NIN,4,SYM,1.0)
IF (LSFR.EQ.1) CALL LRCURV(PRIP,RADIN,NIN,4,SYM,1.0)

C--PLOT INLET ABSOLUTE WHIRL
C
IF (LAMVT.EQ.1) GO TO 80
LRNG = LAMIN(1)
RRNG = LAMIN(1)
DO 70 J=1,NIN
LRNG = AMIN(LRNG,LAMIN(J))
70 RRNG = AMAX1(RRNG,LAMIN(J))
CALL LRANGE(LRNG,RRNG,RRNG,TRNG)
CALL LRLEGN(TITL3,20,0,2.5,0.5,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL17,20,0,4.0,1.3,0.0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,1,0.2,4.2,0.0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1,0.2,4.7,0.0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFIN,LAMIN,NIN,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RADIN,LAMIN,NIN,PLTY,101,PLTX,GRAD)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0.0)
IF (LSFR.EQ.0) CALL LRCURV(LAMIN,SFIN,NIN,4,SYM,1.0)
IF (LSFR.EQ.1) CALL LRCURV(LAMIN,RADIN,NIN,4,SYM,1.0)
GO TO 110
C--PLOT INLET ABSOLUTE TANGENTIAL VELOCITY
C
80 LRNG = VTHIN(1)
RRNG = VTHIN(1)
DO 90 J=1,NIN
LRNG = AMIN(LRNG,VTHIN(J))
90 RRNG = AMAX1(RRNG,VTHIN(J))
CALL LRANGE(LRNG,RRNG,RRNG,TRNG)
CALL LRLEGN(TITL4,36,0,1.1,0.5,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL18,20,0,4.0,1.3,0.0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,1,0.2,4.2,0.0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1,0.2,4.7,0.0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFIN,LAMIN,NIN,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RADIN,LAMIN,NIN,PLTY,101,PLTX,GRAD)
RINSQ = RTIK**2-RPIN**2
DO 100 J=1,101
IF (LSFR.EQ.0) PLTX(J)=PLTX(J)/SORT(RHIN**2+PLTY(J)*RINSQ)
100 IF (LSFR.EQ.1) PLTX(J)=PLTX(J)/PLTY(J)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0.0)
IF (LSFR.EQ.0) CALL LRCURV(VTHIN,SFIN,NIN,4,SYM,1.0)
IF (LSFR.EQ.1) CALL LRCURV(UTHN,RADIN,NIN,4,SYM,1.0)

C---PREPARE FOR PLOTTING OF OUTLET CONDITIONS

110 IF (LSFR.EQ.1) GO TO 130
PLTY(1) = SFOUT(1)
PLTY(101) = SFOUT(NOUT)
DEL = (SFOUT(NOUT)-SFOUT(1))/100.
DC 120 J=2,100
120 PLTY(J) = PLTY(J-1)+DEL
GO TO 150
130 PLTY(1) = RACOUT(1)
PLTY(101) = RADOUT(NOUT)
DEL = (RADOUT(NOUT)-RADOUT(1))/100.
DC 140 J=2,100
140 PLTY(J) = PLTY(J-1)+DEL

C---PLOT OUTLET ABSOLUTE TOTAL PRESSURE

150 IF (LTPL.EQ.1) GO TO 170
LRNG = PROP(1)
RRNG = PROP(1)
DC 160 J=1,NOUT
LRNG = AMIN(LRNG,PROP(J))
160 RRNG = AMAX1(RRNG,PROP(J))
CALL LRANGE(LRNG,RRNG,LRNG,RRNG,TRNC)
CALL LRLEGN(TITL5,32,0,1,5,0,5,0,0)
CALL LRCHS7(12)
CALL LRLEGN(TITL19,20,0,4,0,1,3,0,0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,1,0,2,4,2,0,0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1,0,2,4,7,0,0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFCUT,PRCP,NOUT,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RADOUT,PRCP,NOUT,PLTY,101,PLTX,GRAD)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0,0)
IF (LSFR.EQ.0) CALL LRCURV(PRCP,SFOUT,NOUT,4,SYM,1,0)
IF (LSFR.EQ.1) CALL LRCURV(PROP,RADOUT,NOUT,4,SYM,1,0)
GO TO 190

C---PLOT OUTLET ABSOLUTE TOTAL PRESSURE LOSS

170 LRNG = LOSOUT(1)
RRNG = LOSOUT(1)
DC 180 J=1,NOUT
LRNG = AMIN1(LRNG,LOSOUT(J))
180 RRNG = AMAX1(RRNG,LOSOUT(J))
CALL LRANGE(LRNG,RRNG,LRNG,RRNG,TRNC)
CALL LRLEGN(TITL6,36,0,1,0,0,5,0,0)
CALL LRCHS7(2)
CALL LRLEGN(TITL20,20,0,4,0,1,3,0,0)
IF (LSFR.EQ.0) CALL LRLEGN(TITL13,16,1,0,2,4,2,0,0)
IF (LSFR.EQ.1) CALL LRLEGN(TITL14,8,1,0,2,4,7,0,0)
CALL LRCHS7(4)
IF (LSFR.EQ.0) CALL SPLINT(SFCUT,LOSOUT,NOUT,PLTY,101,PLTX,GRAD)
IF (LSFR.EQ.1) CALL SPLINT(RACOUT,LOSOUT,NOUT,PLTY,101,PLTX,GRAD)
CALL LRCURV(PLTX,PLTY,101,2,SYM,0,0)
IF (LSFR.EQ.0) CALL LRCURV(LOSOUT,SFOUT,NOUT,4,SYM,1,0)
IF (LSFR.EQ.1) CALL LRCURV(LOSOUT,RADOUT,NOUT,4,SYM,1,0)
C--PLOT OUTLET ABSOLUTE WHIRL

150 IF (LAMVT.EQ.1) GO TO 210
   LNRG = LAMOUT(1)
   RRNG = LAMOUT(1)
   DO 200 J=1,NOUT
      LNRG = AMIN1(LNRG,LAMOUT(J))
   200 RRNG = AMAX1(RRNG,LAMOUT(J))
   CALL LRANGE(LNRG,RRNG,LRNG,TRNG)
   CALL LRLEGN(TTTL7,24,0,2.0,0.5,0.0)
   CALL LRCHS7(2)
   CALL LRLEGN(TTTL8,36,0,1.0,0.5,0.0)
   CALL LRCHS7(4)
   IF (LSFR.EQ.0) CALL SPLINT(SFCUT,LAMOUT,NOUT,PLT5,101,PLTX,GRAD)
   IF (LSFR.EQ.1) CALL SPLINT(RACCUT,LAMOUT,NOUT,PLT5,101,PLTX,GRAC)
   CALL LRCURV(PLTX,PLT5,101,2,SYM,0.0)
   IF (LSFR.EQ.0) CALL SPLINT(SFCUT,LAMOUT,NOUT,4,SYM,1.0)
   IF (LSFR.EQ.1) CALL SPLINT(RACCUT,LAMOUT,NOUT,4,SYM,1.0)
   GC TO 240

C--PLOT OUTLET ABSOLUTE TANGENTIAL VELOCITY

210 LNRG = VTHOUT(1)
   RRNG = VTHOUT(1)
   DO 220 J=1,NOUT
      LNRG = AMIN1(LNRG,VTHOUT(J))
   220 RRNG = AMAX1(RRNG,VTHOUT(J))
   CALL LRANGE(LNRG,RRNG,LRNG,TRNG)
   CALL LRLEGN(TTTL8,36,0,1.0,0.5,0.0)
   CALL LRCHS7(4)
   IF (LSFR.EQ.0) CALL SPLINT(SFCUT,LAMOUT,NOUT,101,PLTX,GRAD)
   IF (LSFR.EQ.1) CALL SPLINT(RACCUT,LAMOUT,NOUT,101,PLTX,GRAC)
   CALL LRCURV(PLTX,PLT5,101,2,SYM,0.0)
   IF (LSFR.EQ.0) CALL SPLINT(RACCUT,LAMOUT,NOUT,4,SYM,1.0)
   IF (LSFR.EQ.1) CALL SPLINT(RACCUT,LAMOUT,NOUT,4,SYM,1.0)
   GC TO 240

C--PLOT INPUT BLADE SECTIONS

C--CALCULATE BLADE SECTION PCT CCCRCINATES ALONG PERICIONAL PLANE

240 DC 250 JN=1,NPPPP
   MBL(1,JN) = ZBL(1,JN)
   DO 250 IN=2,NPPPP
      MBL(IN,JN) = MBL(IN-1,JN)+SORT(ZBL(IN,JN)-ZBL(IN-1,JN))**2+
                     RRL(IN,JN)-RRL(IN-1,JN))**2
   250 CONTINUE
   CALL SPLINE(MBL(1,JN),THRL(1,JN),NPPPP,DTCHM,ANG)
   DELRTH = (RRL(1,JN)+RRL(NPPPP,JN))/2.*PITCH
OC 260 IN=1,NPPP
ANG(IN) = ATAN(RBL(IN,JN)*DTHCM(IN))
DELT = THBL(IN,JN)/COS(ANG(IN))
RTHBL(IN,JN) = RFL(IN,JN)+THBL(IN,JN)
SSTHBL(IN,JN) = RTHBL(IN,JN)+DELT/2.

RTHRL(IN,JN) = RTHBL(IN,JN)-DELT/2.
RRTTHRL(IN,JN) = RTHBL(IN,JN)+DELTTH
SSTPBL(IN,JN) = SSTHBL(IN,JN)+DELTTH

C--CALCULATE RANGE OF PLOTS, AND SET UP FOR PLOTTING INDIVIDUAL
C--PLACE SECTIONS

LRNG = MBL(1,1)
RRNG = MBL(NPPP,1)
BRNG = PTHBL(1,1)
TRNG = SSTHBL(NPPP,NBLPL)
DO 270 JN=1,NBLPL
LRNG = AMINI(LRNG,MBL(1,JN))
RRNG = AMAXI(RRNG,MBL(NPPP,JN))
DO 270 IN=1,NPPP
BRNG = AMINI(BRNG,PTHBL(IN,JN))

TRNG = AMAXI(TRNG,SSTHBL(IN,JN))
RRTEM = RRNG
DELLR = RRNG-LRNG
DELAT = TRNG-BRNG

DELNG = AMAXI(DELLR,DELAT)
RRNG = LRNG+DELLNG
TRNG = RRNG+DELNG

CALL LRANGE(LRNG,RRNG,BRNG,TRNG)

C--PLACE SECTIONS AND SHOW SOLIDITY
CALL LRLEGN(TITL10,52,0,2.7,0.7,0.0)
DO 280 JN=1,NALPL
CALL LRCHS7(3)
CALL LRCHSV(TITL11(6),1,4,0)
CALL LRLEGN(TITL11,24,0,3,0,9.5,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL24,40,0,2,8,1,3,0.0)
CALL LRLEGN(TITL25,36,1.0,2,3,3,0.0)
CALL LRCHS7(4)
CALL LRCURV(MBL(1,JN),RTHBL(1,JN),NPPP,2,SYM,0.0)
CALL LRCURV(MBL(1,JN),RTHRL(1,JN),NPPP,4,SYM,0.0)
CALL LRCURV(MBL(1,JN),STHBL(1,JN),NPPP,2,SYM,0.0)
CALL LRCURV(MBL(1,JN),PTHBL(1,JN),NPPP,2,SYM,0.0)
CALL LRCURV(MBL(1,JN),RRTTHRL(1,JN),NPPP,2,SYM,2,SYM,0.0)
CALL LRCURV(MBL(1,JN),RRTHBL(1,JN),NPPP,4,SYM,0.0)
CALL LRCURV(MBL(1,JN),SSTHBL(1,JN),NPPP,2,SYM,0.0)
CALL LRCURV(MBL(1,JN),RRTTHBL(1,JN),NPPP,2,SYM,2,SYM,0.0)

280 CALL LRCURV(MBL(1,JN),PPTHBL(1,JN),NPPP,2,SYM,1.0)
C--CALCULATE RANGE OF PLCT, AND SET UP FOR PLOT OF MULTIPLE
C--PLACE SECTIONS

RRNG = RRTEM
TRNG = SSTHBL(NPPP,NBLPL)
DO 290 JN=1,NBLPL
DC 290 IN=1,NPPP

TRNG = AMAXI(TRNG,SSTHBL(IN,JN))
DELAT = TRNG-RRNG
DELNG = AMAXI(DELLR,DELAT)
RRNG = LRNG+DELNG
TRNG = RRNG+DELNG

CALL LRANGE(LRNG,RRNG,BRNG,TRNG)

C--PLOT MULTIPLE BLADE SECTIONS
CALL LRGRID(3,3,11,0,11.0)
CALL LRCHS7(3)
CALL LRLECN(TITL12,24.0,3.4,9.5,0.0,0)
CALL LRCHS7(2)
CALL LRLECN(TITL24,40.0,2.8,1.3,0.0)
CALL LRLECN(TITL25,36.1,0.2,3.3,0.0)
CALL LRCHS7(4)
EC = 0.0
DO 300 JN=1,NBLPL
IF (JN.EQ.NBLPL) EC = 1.0
CALL LRCIRV(PPL(1,JN),RTHPL(1,JN),NPPP,2,SYM,0.0)
CALL LRCURIV(RMBl(1,JN),RTHBL(1,JN),NPPP,4,SYM,0.0)
CALL LRCURIV(MPL(1,JN),STHL(1,JN),NPPP,2,SYM,EC)
300 CALL LRCURIV(ZRL,RBL,SYM,1.0)
RETURN
END

SUBROUTINE MESH0
C
C--MESH0 CALCULATES COORDINATES OF AN CRTF-Cgononal MESH
C--COVERING THE SOLUTION REGION
C
COMMON/INPUT/GAM,AR,MSFL,OMEGA,RECFAC,VELTOL,FNEW,DNEW,MBI,MBO,
MM,MHT,NBL,NHUB,NTP,NIN,NOUT,NBLPL,NPPP,KSTAT,NST,LSFR,
LTPL,LAMV,IMESH,ISL,INE,ISTATL,IPLOT,ISUP,ISTR,IDEBUG,
70MIN,70M,70MOD,70MOUT,70IN,70OUT,70CUT,70HUB(50),
RHub(50),RTip(50),SFIn(50),RAd(50),TIP(50),Prip(50),
LAMIN(50),VThin(50),SFOut(50),RAc(50),PROP(50),LOSO(50),
LAMOUT(50),VHOut(50),ZHST(50),ZTST(50),FLFR(50),
ZAL(50,50),RBL(50,50),TBBl(50,50),TNBL(50,50)
COMMON/CAI/CCTION/MM1,MHTPI,CP,EXPON,TCROG,PITCH,CURVH,CURVI,
CURVHO,CURVTO,HRN,RTIN,RTOUT,RELH,REL,RTET,
ZLE(50),ZLE5(50),ZTE(50),RTE(50),ZLEom(101),RLEcm(101),
SLEom(101),THEOm(101),ZTHEm(101),RTHEm(101),STHEm(101),
THTEom(101),THE(101),ITE(101),ZOM(100,101),ROM(100,101),
SCH(100,101),SOM(100,101),RTH(100,101),DTH(100,101),
DTH(100,101),FLOSS(100,101),CPHI(100,101),SPHI(100,101)
COMMON/CRSCM/RTMP(100),SDRV(100),REFZ,REFR,REFSL,MARK
DIMENSION RRAD(100,101),SOM(100),AAA(100)
EXTERNAL CRSCC
MARK = 1
CALL CRSCC(7CM(1,1),RMR,SL1)
MARK = 0
C
C--STORE RRAD ON HUB CONTOUR
DO 10 I=1,NHUB
10 RRAD(I,1) = RHub(I)
C
C--CALCULATE RRAD ON SHROUD CONTOUR
CALL SPLINT(ZTip,RTip,NTip,ZHub,NHub,RRAD(1,MHTPl),AAA)
C
C--COMPUTE RRAD ON RADIAL LINES FROM HUB TO TIP
DC 20 I=1,NHUB
DELR = (RRAD(I,MHTPl)-RRAD(I,1))/FLCAT(MHT)
DO 20 J=2,MHT
20 RRAD(I,J)= RRAD(I,J-1)+CELR
C
C--COMPLETE ZOM ON HUB
MEIM1 = MBI-1
DELZ = (ZOMRI-ZCMIN)/FLCAT(MBI+1)
ZOM(I,1)= ZOMIN
DC 30 I=2,MBI
30 ZCM(I,1)= ZOM(I-1,1)+DELZ
DELZ = (ZOMPI-ZMI)/FLCAT(MBI-MBI)
MPPI1 = MBI+1
DO 40 I=MPIP1,MBI
40 ZOM(I,1)= ZOM(I-1,1)+DELZ
DELZ = (ZOMCUT-ZOMBO)/FLCAT(MMI-MBI)
MBOPI = MBO+1
DO 50 I=MBOPI,MM
50 ZCM(I,1)= ZOM(I-1,1)+DELZ

C
C--COMPUTE ENTRIES TO ZOM AND ROM ROW BY ROW FROM HUB TO SHROUD
C
DO 150 J=2,MHTPI
C
C--CALCULATE R-COORDINATES, SLOPES, AND ANGLES OF PREVIOUS ROW
CALL SPLINT(ZHUB,RRAD(I,J-1),NHUR,ZCM(I,J-1),MM,ROM(I,J-1),SLOM)
DO 60 I=1,MM
CPHI(I,J-1) = 1./SQRT(1.+SLOM(I)**2)
60 SPHI(I,J-1) = SLOM(I)*CPHI(I,J-1)
C
C--CALCULATE RTMP, AND SECOND DERIVATIVES, SDRIV, ON PRESENT ROW.
DO 70 I=1,NHUR
70 RTMP(I)= RRAD(I,J)
CALL SPLINE(ZHUB,RTMP,NHUR,AAA,SDRIV)
C
C--MOVE ALONG PRESENT ROW, ONE POINT AT A TIME, LOCATING ZCM
C--COORDINATES OF ORTHOGONAL MESH POINTS ALONG THE ROW.
DC 140 I=1,MM
REF7= ZOM(I,J-1)
REFR= ROM(I,J-1)
DC 80 K=2,NHUR
IF (ZHUB(K).LT.ZOM(I,J-1)) GO TO 80
DELR = RRAD(K-1,J)-RRAD(K-1,J-1)-(ZOM(I,J-1)-ZHUB(K-1)) /
1(ZHUB(K)-ZHUB(K-1))*(RRAO(K-1,J)-RRAD(K-1,J-1)-PRAC(K,J)+
2RRAD(K,J-1))
GC TC 90
80 CONTINUE
90 IF (ABS(SLOM(I)).LE.0.0001) GO TO 120
C
C--LOCATE INTERSECTION CF LINE NORMAL TO PREVIOUS ROW WITH PRESENT ROW
REFSL= -1./SLOM(I)
DELZ= DELR/REFSL
IF (ABS(REFSL).GT.10.) DELZ=DELR/SIGN(10.,REFSL)
IF (REFSL.LT.0.) GO TO 100
A= ZCM(I,J-1)
B= A+2.0*DELZ
GC TC 110
100 A= ZCM(I,J-1)
B= A+2.0*DELZ
110 TCLER= DELR/110.
IF(ABS(SLOM(I)).LE.0.01) TOLER = TOLER/ABS(SLOM(I))*01
131
CALL ROOT(A,R,0.,CROSCD,TOLER,71,SL1)
   R1 = REFR+REFSL*(Z1-REFZ)
   GO TO 130
C
C--LCCATE INTERSECTION IF NORMAL TO PREVIOUS ROW IS RACIAL
120 REFSL= 0.
   CALL CROSCD(7OM(I,J-1),RMR,SL1)
   71= ZCM(I,J-1)
   R1= RMR
C
C--CALCULATE FINAL LOCATION OF ZOM
130 Z2= (7OM(I,J-1)+(RCM(I,J-1)-R1)*SL1+Z1*SL1**2)/(1.+SL1**2)
   R2= R1+SL1*(Z2-Z1)
   140 ZCM(I,J)= (Z1+Z2)/2.
   150 CONTINUE
C
C--CALCULATE R-COORDINATES, SLOPES, AND ANGLES OF FINAL ROW
C-- FOR THE GIVEN VALUE OF Z
C--CALL SPLIT(ZHUB,RRAD(I,MHTPI),NHUB,ZCM(I,MHTPI),MM,
C--ROM(I,MHTPI),SLOM)
DC 160 I=1,MM
   CPHI(I,MHTPI) = 1./SQRT(SLOM(I)**2+1.)
   SPHI(I,MHTPI) = SLOM(I)*CPHI(I,MHTPI)
RETURN
END

SUBROUTINE CROSCD(Z,RMR,SL1)

C--CROSCD CALCULATES R AS A FUNCTION OF Z ALONG A CURVE AND ITS
C--INTERSECTING STRAIGHT LINE, AND COMPUTES THE DIFFERENCE BETWEEN THE
C--VALUES OF R ON THE STRAIGHT LINE AND CURVE FOR A GIVEN VALUE OF Z.

C
COMMON/INPUTT/CAI,AR,MSFL,OMEGA,REDFAC,VELTOL,FNEW,DNEW,MBI,WBO,
1 MM,MHT,NL,NHUB,NTIP,NIN,NOUT,NRLPL,NPPP,NOSTAT,NSL,LSFR,
2 LTPL,LAIVT,IMESH,ISLINE,ISTAT,ITPLCT,ITSC,ITDEBUG,
3 ZMIN,ZOMBI,ZOMBO,ZOMOUT,ZHIN,ZTIN,ZOUT,ZHUR,ZHUB(50),
4 RHUB(50),TIP(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAND(50),VTHIN(50),SFOUT(50),RADOUT(50),PREP(50),LOSOUT(50),
6 LAND(50),VTHOUT(50),ZHST(50),ZTST(50),FLFR(50),
7 ZRRL(50),ZRL(50),UBL(50),ZNL(50),THBL(50),TNBL(50),
COMMON/CROSM/RMP(100),SDRIV(100),REFZ,REFR,REFSL,MARK
IF(MARK.EQ.1) RETURN
C
C--LCCATE POSITION OF Z IN ZHUB ARRAY
DO 10 I=1,NHUB
   IF (Z.LE.ZHUB(I)) GO TO 20
10 CONTINUE
C
C--COMPUTE R-COORDINATES (RCURV) AND SLOPE (SL1) ON THE CURVE
C--FOR THE GIVEN VALUE OF Z.
20 DEL7 = ZHUB(I)-ZHUB(I-1)
   SCI = SDRIV(I)
   SCI1 = SDRIV(I-1)
   ZM7 = ZHUB(I)-Z
   ZM7H = Z-ZHUB(I-1)
   RT7 = RMP(I)/DELZ
\[ \begin{align*}
& n_{T11} = \frac{RT_{MP}(I-I)}{\delta_{7}} \\
& R_{CURV} = \frac{SD_{11} + Z_{M} Z_{H}^{*} 3}{6} / \delta_{7} + SD_{I} * Z_{M} Z_{H}^{*} 3 / 6 + (RT_{I} - SD_{I} * \delta_{7} / 6) \\
& \times 1 * Z_{M} Z_{H} + R_{T11} - SD_{I} * \delta_{7} / 6. \\
& \delta_{7} = -SD_{I} * Z_{M} Z_{H}^{*} 2 / 2 / \delta_{7} + SD_{I} * Z_{M} Z_{H}^{*} 2 / 2 / \delta_{7} + RT_{I} - RT_{T11} - 1 \quad (SD_{I} - SD_{11}) * \delta_{7} / 6. \\
& \text{IF} (REFS_{L} \cdot E_{C} \cdot 0.) \text{ GC TO 30}
\end{align*} \]

C --- COMPUTE R-COORDINATE (RLINE) ON STRAIGHT LINE
C --- FOR GIVEN VALUE OF Z
RLINE = \text{REFR} + \text{REFSL} * (Z - \text{REFZ})

C --- COMPUTE DIFFERENCE IN R-COORDINATES
RMR = R_{CURV} - RLINE
RETURN

C --- SPECIAL CASE FOR RACIAL STRAIGHT LINE
30 RMR = R_{CURV}
RETURN
END

SUBROUTINE PRECAL
C --- PRECAL CALCULATES MANY OF THE REQUIRED FIXED CONSTANTS
C
COMM/INPUT/GAM, AR, MSFL, LAMDA, REDFAC, VELTOL, FNW, DNEW, MBI, MB0,
1 MM, MHT, NBL, NHU, NTIP, NIN, NOUT, NBLPL, NPP, NOSTAT, NSL, LSF,
2 LTPL, LAMVT, IMESH, ISLNE, STATL, IPLCT, ISUPER, ITSCH, IDEBUG,
3 ZOMIN, ZOMB, ZOMOUT, ZMIN, ZITIN, ZOUT, ZTCUT, ZHUB (50),
4 RHHU (50), RTHI (50), RTIP (50), SFIN (50), RADI (50), TIP (50), PRIP (50),
5 LAMIN (50), VTHIN (50), LAMOUT (50), RADOUT (50), PRCP (50), LSCOUT (50),
6 LAMCUT (50), VTHOUT (50), ZHST (50), ZTST (50), FLFR (50),
7 RL (50, 50), RRL (50, 50), TRL (50, 50), TNL (50, 50)
COMMON/CALCON/MM1, MHTP1, CP, EXPON, TGRAC, PITCH, CURVHI, CURVTI,
1 CURVHI, CURVTH, RHIN, RTIN, RHOUT, RTOUT, RLEH, RLET, RTET, RTE,
2 ZLEH (50), ZLEE (50), ZTE (50), ZTEH (50), ZLECM (101), RLECM (101),
3 SLECM (101), TLECM (101), ZTEOM (101), RTEOM (101), STEOM (101),
4 THTOM (101), ILE (101), ITE (101), ZOM (100, 101), RCM (100, 101),
5 SOM (100, 101), TOM (100, 101), BTH (100, 101), DTHDS (100, 101),
6 DTHDT (100, 101), PLOSS (100, 101), CPH (100, 101), SPHI (100, 101)
DIMENSION DDX (100), DDX2 (100), TEMPO (100), RELTCP (50)
REAL MSFL, LAMDA, LAMOUT, LAMDAI, LOUT
C
C --- INITIALIZE TIPE, RHCIPF, LAMDAF, RHOOPF, AND RVTHTA
C
CALL LAMNIT
CALL RHTNIT
CALL TIPNIT
CALL RHINT
C --- CALCULATE PROP, IF LCSOUT WAS GIVEN AS INPUT
IF (LTPL.EQ.0.0) GO TO 20
DC 10 J=1,NCUT
TINP = TIPE(SFOUT(J))
LAMDAI = LAMDAF(SFOUT(J), 1, 1)
TCP = TOPF(SFOUT(J))
PRINP = RHOFP(SFOUT(J)) * AR * TINP

133
RFLTCPI(J) = 1.0 - LOSOUT(J)
10 PROP(J) = RFLTCPI(J) * PRINP*(TOP/TINP)**(GAM*EXPON)
20 CALL RHONIT

C--INITIALIZE THE BTH ARRAY
C
DO 30 J=1,MHTPI
DO 30 I=1,MM
30 RTH(I,J) = PITCH

C--INITIALIZE THE FLFR ARRAY IF IT WAS NOT READ IN
C
IF (NSL.GE.1) GO TO 50
NSL = 11
FLFR(1) = 0.0
FLFR(11) = 1.0
DO 40 J=2,10
40 FLFR(J) = FLFR(J-1)+0.1
GO TO 80

C--SET END POINTS FOR FLFR ARRAY
C
50 IF (FLFR(1).EQ.0.0) GO TO 70
TEMP1 = 0.0
DO 60 JL=1,NSL
TEMP2 = FLFR(JL)
FLFR(JL) = TEMP1
60 TEMP1 = TEMP2
NSL = NSL+1
FLFR(NSL) = TEMP1
70 IF (FLFR(NSL).EQ.1.0) GO TO 80
NSL = NSL+1
FLFR(NSL) = 1.0

C--CALCULATE SOM FROM THE ZOM, ROM ARRAYS
C
DO 90 J=1,MHTPI
SOM(1,J) = 0.0
DO 90 I=2,MM
90 SOM(I,J) = SOM(I-1,J)+SORT((ZOM(I,J)-ZCM(I-1,J))**2+(ROM(I,J)-
1RCM(I-1,J))**2)

C--CALCULATE TOM FROM THE ZOM, ROM ARRAYS
C
DO 100 I=1,MM
TOM(I,1) = 0.0
DO 100 J=2,MHTPI
100 TOM(I,J) = TOM(I,J-1)+SORT((ZOM(I,J)-ZCM(I,J-1))**2+(ROM(I,J)-
1ROM(I,J-1))**2)

C--CALCULATE CURVATURES ON HUB AND SP-RCUD
C--AT ENTS OF ORTHOGONAL MESH
C
CALL SPLINE(ZCM(1,1),ROM(1,1),MM,CYCX,CYCX2)
CURVHI= DYDX2(1)/(1.+DYDX(1)**2)**1.5
CURVH0= CYCX2(MM)/(1.+DYDX(MM)**2)**1.5
CALL SPLINE(ZCM(1,MHTPI),ROM(1,MHTPI),MM,CYCX,CYCX2)
CURVTI= DYDX2(1)/(1.+DYDX(1)**2)**1.5
CURVT0= CYCX2(MM)/(1.+DYDX(MM)**2)**1.5

134
C--CALCULATE LEADING EDGE ARRAY, ZLE, RLE, FROM ZBL AND RBL ARRAYS
C--CALCULATE INTERSECTION OF LEADING EDGE WITH HUB AND SHROUD PROFILES
C
DO 110 JN=1,NRLPL
  ZLE(JN) = ZBL(JN)
110 RLE(JN) = RBL(JN)
CALL INRSCT(ZHUB, RHUB, NHUB, ZLE, RLE, NBLPL, ZLEH, RLEH)
CALL INRSCT(ZTIP, RTIP, NTIP, ZLE, RLE, NBLPL, ZLET, RLET)
C
C--CALCULATE TRAILING EDGE ARRAY, ZTE, RTE, FROM ZBL AND RBL ARRAYS
C--CALCULATE INTERSECTIONS OF TRAILING EDGE WITH HUB AND SHROUD PROFILES
C
DO 120 JN=1,NRLPL
  ZTE(JN) = ZBL(NPPP,JN)
120 RTE(JN) = RBL(NPPP,JN)
CALL INRSCT(ZHUB, RHUB, NHUB, ZTE, RTE, NBLPL, ZTEH, RTEH)
CALL INRSCT(ZTIP, RTIP, NTIP, ZTE, RTE, NBLPL, ZTEH, RTEH)
C
C--CALCULATE ORTHOGONAL MESH ARRAYS AT THE LEADING EDGE
C--ZLEOM, RLEOM, SLEOM, THLEOM
C--CALCULATE ILE ARRAY OF MESH POINT LOCATIONS INSIDE BLADE
C--LEADING EDGE
C
  ZLEOM(1) = ZLEH
  RLEOM(1) = RLEH
  ZLECM(MHTPI) = ZLET
  RLEOM(MHTPI) = RLET
  DC 130 J=2,MHT
130 CALL INRSCT(ZOM(1,J), RCM(1,J), WM, ZLE, RLE, NBLPL, ZLEOM(J), RLEOM(J))
  DO 160 J=1,MHTPI
  DO 140 I=1,WM
     IF (ZLEOM(J).LE.ZOM(I,J)) GO TO 150
140 CONTINUE
150 ILE(J) = I
     ILE(J) = I-1
160 SLEOM(J) = SOM(ILEJ,J)+SORT((ZLEOM(J)-ZOM(ILEJ,J))**2+(RLEOM(J)-
     RCM(ILEJ,J))**2)
  DO 170 JN=1,NRLPL
170 TTEM(JN) = THPL(JN)
  CALL SPLINT(RLE, TTEM, NBLPL, RLECM, MHTPI, THLEOM, CYCX)
C
C--CALCULATE ORTHOGONAL MESH ARRAYS AT THE TRAILING EDGE:
C--ZTEOM, RTEOM, STEOM, THTEOM
C--CALCULATE LTE ARRAY OF MESH POINT LOCATIONS INSIDE BLADE
C--TRAILING EDGE
C
  ZTEOM(1) = ZTEH
  RTEOM(1) = RTEH
  ZTECM(MHTPI) = ZTET
  RTEOM(MHTPI) = RTET
  DC 180 J=2,MHT
180 CALL INRSCT(ZOM(1,J), RCM(1,J), WM, ZTE, RTE, NBLPL, ZTEOM(J), RTEOM(J))
  DO 210 J=1,MHTPI
  ILEJ = ILE(J)-1
  DO 190 I=1,ILEJ,WM
     IF (ZTEOM(J).LT.ZOM(I,J)) GO TO 200
190 CONTINUE
190 ILE(J) = I-1
200 ILE(J) = I-1
LIBRARY = I-1

210 TTEM(J) = SOM(ITEJ,J) + SCRT((2TTEM(J) - TTEM(ITEJ,J))**2 + RTEM(J)**2)

DO 220 JN=1,NRLPL
   CALL SPLINT(RTE,TTEM,NRLPL,RTEM,MHTPI,THTEM,ICYC)

C--CALCULATE THE ETA GRADIENTS ON THE ORTHOGONAL MESH
C
CALL THEOM
C
C--CORRECT RTH FOR BLADE THICKNESS ON THE ORTHOGONAL MESH
C
CALL THIKOM
C
C--CALCULATE ACTUAL-TO-IDEAL RELATIVE TOTAL PRESSURE RATIOS
C--DOWNSTREAM OF BLADE, AND CALCULATE LOSS ON ORTHOGONAL MESH
C
C--CORRECT RTH FOR TOTAL PRESSURE LOSS
C
CALL LOSSOM
C
C--REDUCE MASSFLOW, WHEEL SPEED, AND WHIRL FOR REDUCED FLOW SOLUTION
C
IF (REDFAC.EQ.1.0) RETURN
OMEGA = OMEGA*REDFAC
MSFL = MSFL*REDFAC
DO 230 J=1,NIN
   LAMIN(J) = LAMIN(J)*REDFAC
230 VTHIN(J) = VTHIN(J)*REDFAC

DO 240 J=1,NOUT
   LAMOUT(J) = LAMOUT(J)*REDFAC
240 VTHOUT(J) = VTHOUT(J)*REDFAC

C--RE-INITIALIZE LAMDAF AND RVTHTA FOR REDUCED FLOW
C
CALL LAMNIT
CALL RVTNIT
RETURN END

SUBROUTINE THEOM
C
C--THEOM CALCULATES THE DERIVATIVES OF ETA WITH RESPECT TO S AND T
C--DIRECTIONS ON THE ORTHOGONAL MESH
C
COMMON/INPUTT/GAM,AR,MSFL,OMEGA,REDFAC,VELTCL,FNEW,DNEW,MBI,MBO, 1 MM,MHT,NRL,NHUR,NTP,NIN,NOUT,NBLPL,NPPP,NCSTAT,NSL,LSFR,
2 LTPL,LAMVT,IMESH,ISLINE,ISTATL,IPLOT,ISSUPER,ISOSM,IDEBUG,
3 ZMIN,ZOMBI,ZCMBO,ZOMOUT,ZHIN,ZHOUT,ZTCUT,213
4 RHUR(50),ZTIP(50),Rtip(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTHIN(50),SOUT(50),RACOUT(50),RACOUT(50),PROP(50),LOSOUT(50),
6 LAMOUT(50),VTHOUT(50),ZHST(50),ZTST(50),FLFR(50),
7 ZBL(50,50),RBL(50,50),THRL(50),TNBL(50,50)
COMMON/CCALCC/MMM1,MHTPI,CP,EXPN,TGROG,PITCH,CURVHI,CURVTI,
1 CURVHO,CURVTO,RHIN,RTIN,RHOUT,RTOUT,RLEH,RLET,RTHE,RTET,
C--CALCULATE GRADIENTS OF THETA WITH RESPECT TO DISTANCE ALONG INPUT C--ZPC, RPC LINES
C-- LOCATE INTERSECTIONS OF INPUT ZBL, RBL LINES WITH LINES FROM C-- HUB TO TIP AT TEN PERCENT CHORD INTERVALS
DC 50 JN=1, NBLPL
  DELZ = 0.1*(ZRL(NPPP,JN)-ZBL(1,JN))
  ZPC(1,JN) = ZRL(1,JN)
DC 10 KN=2, 11
  ZPC(KN,JN) = ZPC(KN-1,JN)+DELZ
C--CALCULATE R COORDINATES AND ANGLES WITH RESPECT TO Z AXIS AT C--INTERSECTION POINTS
CALL SPLINT(ZBL(1,JN), RBL(1,JN), NPPP, ZPC(1,JN), 11, RPC(1,JN),
  IANGZ(1,JN))
DC 20 KN=1, 11
  IANGZ(KN,JN) = ATAN(IANGZ(KN,JN))
C--CALCULATE ARC LENGTH ALONG INPUT LINES USING INPUT POINTS
  S7RBL(1) = 0.
DO 30 IN=2, NPPP
  30 S7RBL(IN) = S7RBL(IN-1)+SQR((ZBL(IN,JN)-ZBL(IN-1,JN))**2
      + (RBL(IN,JN)-RBL(IN-1,JN))**2)
C--CALCULATE ARC LENGTH ALONG INPUT LINES USING POINTS AT TEN C--PERCENT OF CHORD
  S7RPC(1) = 0.
DO 40 KN=2, 11
  40 S7RPC(KN) = S7RPC(KN-1)+SQR((RPC(KN,JN)-RPC(KN-1,JN))**2
      + (ZPC(KN,JN)-ZPC(KN-1,JN))**2)
C--CALCULATE THETA AND CHANGE OF THETA WITH ARC LENGTH ALONG INPUT LINES
  50 CALL SPLINT(S7RBL, THBL(1,JN), NPPP, S7RPC, 11, THPC(1,JN), DTHDSP(1,JN)
C--CALCULATE GRADIENT OF THETA WITH RESPECT TO DISTANCE UP TEN PERCENT C--CHORD ZPC, RPC LINES
C
DC 80 KN=1, 11
C--CALCULATE SLOPES AND ANGLES WITH RESPECT TO R AXIS ALONG THE C--TEN PERCENT CHORD, HUB-TIP LINES
DC 60 JN=1, NRLPL
  ZTEM(JN) = ZPC(KN,JN)
  RTEM(JN) = RPC(KN,JN)
60 TTEM(JN) = THPC(KN,JN)
CALL SPLINE(RTEM, ZTEM, NRLPL, DYDX, DYDX2)
  S7RPC(1) = 0.
  AGR(KN,1) = ATAN(DYDX(1))
DO 70 JN=2, NRLPL
  70 S7RPC(JN) = S7RPC(JN-1)+SQR((RPC(KN,JN)-RPC(KN,JN-1))**2
      + (ZPC(KN,JN)-ZPC(KN,JN-1))**2)
  AGR(KN,JN) = ATAN(DYDX(JN))
C--CALCULATE CHANGE OF THETA WITH ARC LENGTH ALONG HUB-TIP LINES
CALL SPLINE(SZRPC,TTEMP,NRLPL, CSX, DITY2)
DO 40 JN=1,NRLPL
40 DTHDTP(KN,JN)= DITY(XJN)
C
C--CALCULATE DTHDZ AND DTHOR FROM DTHOSP AND DTHDTP AT POINTS OF
C--INTERSECTION OF INPUT LINES AND HUB-TIP LINES
C
DO 90 JN=1,NRLPL
DO 90 KN=1,11
CFSAB= COS(ANG(7,KN,JN)+ANGR(KN,JN))
DTHDZ(KN,JN)= (DTHDTP(KN,JN)*COS(ANG(KN,JN)))-CTHC(TP(KN,JN)*SIN(L8N817,KN,JN)))/COSAB
90 DTHDZ(KN,JN)= (-CTHCSP(KN,JN)*SIN(ANGR(KN,JN)))+DTHDTP(KN,JN)*COS(1ANGX(KN,JN)))/COSAB
C
C--INTERPOLATE TO OBTAIN DTHDZ AND DTHOR AT THE POINTS OF THE CRTHCGONAL
C--MESH
C--ROTATE DTHDZM AND DTHDM ON ORTHOGONAL MESH TO OBTAIN DTHOS AND DTHOT
C--THE GRADIENTS OF THETA IN THE S AND T DIRECTIONS
C
II = 1
JJ = 1
DC 100 J=1,MHTP1
ILEJ = ILE(J)
ITEJ = ITE(J)
DC 100 I=ILEJ,ITEJ
CALL LININT(ZPC,RPC,DTHDZII,NRLPL,11,50, ZOM(I,J), ROM(I,J),
1 DTDZOM,II, JJ)
CALL LININT(ZPC,RPC,DTHCRII,NRLPL,11,50, ZOM(I,J), ROM(I,J),
1 DTDROM,II, JJ)
DTHDZ(S,J) = DTDZOM*GPHI(S,J)+DTHDRM*SPHI(S,J)
100 DTHOT(S,J) = DTHDZOM*GPHI(S,J)+DTHDM*SPHI(S,J)
RETURN
END

SUBROUTINE THIKOM
C
C--THIKOM CALCULATES THE BLADE THICKNESS IN THE THETA DIRECTION AT
C--THE POINTS OF THE ORTHOGONAL MESH
C
COMMON/INPUTT/CAM,AR,MSL,OMEGA,REDFAC, VELTOL, FNEW, DNEW, MBI, MBC,
1 MM, MHT, NBL, NHRU, NTIP, NIN, NOIT, NRLPL, NPPP, NOSTAT, NSL, LSF, R,
2 LTHPL, LAMVT, IMES, ISPLINE, ISTATL, IFLCT, IUMEC, IDGEUE,
3 ZOMIII, ZOMBI, ZOMBO, ZOMOUT, ZINT, ZTIN, ZTOUT, ZTCUT, ZTHUB(50),
4 RHUR(50), RTIP(50), RTP(50), SFIN(50), RADIN(50), TIP(50), PRP(50),
5 LAMIN(50), VTHIN(50), SFOUT(50), RAOUT(50), PRCP(50), LSCUT(50),
6 LAMOUT(50), VTHOUT(50), ZHT(50), ZTST(50), FLFRT(50),
7 ZBL(50,50), RBL(50,50), THBL(50,50), TNL(50,50),
COMMON/CALCON/MMM1, MHTPI, CP, EXPON, TGRG, PITCH, CURVHI, CURVT,
1 CURVHO, CURVTO, RHIN, RHTIN, RHCUT, RTOUT, RHLE, RLET, RTH, RLET,
2 ZLE(50), RLE(50), TLE(50), LTE(50), LEOM(101), LECM(101),
3 SLEOM(101), TLEOM(101), ITICE(101), RTEDM(101), SEDOM(101),
4 THEDOM(101), TIE(101), IOM(100,101), ROM(100,101),
5 SOM(100,101), TCM(100,101), BTH(100,101), DTM(100,101),
6 DTHD(100,101), PLOSM(100,101), GPHI(100,101), SPHI(100,101),
138
DIMENSION DIST(50), OTOS(50), ANG(50), CRTHBL(50, 50)

C--CALCULATE BLADE THICKNESS IN THE THETA DIRECTION FROM INPUT
C--THICKNESSES NORMAL TC MEAN CAMBER LINE

DO 20 JN=1,NBLPL
  DIST(JN) = 0.
DO 10 IN=2,NPPP
  10 DIST(IN) = DIST(IN-1) + SORT((ZBL(IN, JN)-ZBL(IN-1, JN))**2 +
     (RBL(IN, JN)-RBL(IN-1, JN))**2)
     CALL SPLINE(DIST, THBL(1, JN), NPPP, DTDS, ANG)
  CALL SPLINE(DIST, THBL(IN, JN), NPPP, DTDS, ANG)
  ANG(IN) = ATAN(RBL(IN, JN)-RBL(IN-1, JN))
  DC 20 IN=1,NPPP
  20 DRTHBL(IN, JN) = TNL(BL(IN, JN), COS(ANG(IN))

C--INTERPOLATE TO OBTAIN BLADE THICKNESS IN THETA DIRECTION AT THE
C--POINTS OF THE CRTHGONAL MESH

II = 1
JJ = 1
DC 30 J=1,NHTP1
  ILEJ = ILE(J)
  ITEJ = ITE(J)
DC 30 I=ILEJ,ITEJ
  CALL LINT(NZL, RRL, DRTHBL, NPPP, NBLPL, 50, 50, 7CM(I, J), ROM(I, J),
     lDRTH, II, JJ)
  BTH(I, J) = BTH(I, J) + DRTH/ROM(I, J)
30 CONTINUE
RETURN
END

SUBROUTINE LOSSOM

C--LOSSOM COMPUTES THE RATIO OF ACTUAL TO IDEAL RELATIVE TOTAL PRESSURE
C--DOWNSTREAM OF THE BLADE, AND THEN DISTRIBUTES THIS LOSS LINEARLY
C--THROUGH THE BLADE ROW FROM TRAILING TO LEADING EDGE BY ADDING
C--TO THE BLADE THICKNESS AT THE CRTHGONAL MESH POINTS

COMMON SRW, SR E, ITER, IENC, NREAD, NWRIT
COMMON INPUT/GAM, AR, MSFL, OMEGA, REDFAC, VELTOL, FNEW, DNEW, MRE, MBO,
  ~MM, MHT, NRT, NHUR, NTIP, NIN, NOUT, NBLPL, NPPP, NSTAT, NSL, LSNR,
  LTPL, LAMVT, IMSH, ISSL, ISSLT, ISSL, ISUPER, ITSON, IDEBUG,
  ZMIN, ZMBI, ZMBO, ZMOUT, ZHIN, ZTIN, ZHOUT, ZTOUT, ZMB(50),
  RHUR(50), ZTIP(50), RTIP(50), SFON(50), RADIN(50), TIP(50), PRIP(50),
  LAMIN(50), VTHIN(50), SFOUT(50), RADOUT(50), PROP(50), LOSOUT(50),
  LAMOUT(50), VTHOUT(50), ZHST(50), ZTST(50), FFR(50),
  ZBL(50, 50), RBL(50, 50), THBL(50, 50), TNLB(50, 50)
COMMON/CALCON/MWI, MHTPL, CP, EXPOS, TGRG, PITCH, CURVHI, CURVTI,
  CURVHO, CURVTO, RHIN, RTIN, RHOUT, RTOUT, RLEH, RLET, RTHET, RTHETE,
  ZLE(50), ZLE(50), ZTE(50), ZTE(50), ZLEOM(101), ZLEOM(101),
  SLFOM(101), SLFOM(101), SLOM(101), SLOM(101),
  THTOM(101), THTOM(101), THOM(101), THOM(101),
  SCM(100, 101), TCM(100, 101), ETH(100, 101), ETH(100, 101),
  DTHOM(100, 101), DTHOM(100, 101), CPHI(100, 101), CPHI(100, 101),
  DIMENSION RELTOP(50), SF(101), GRADI(101), PRAI(101),
REAL LAMDAF, LAMIN, LAMOUT, LAMCAI, LOSOUT
C--CALCULATE ACTUAL-TO-Ideal RELATIVE TOTAL PRESSURE RATIO
C--IN DOWNSTREAM INPUT PERTINENT
C
DC 20 J=1,NCUT
TINP = TIPF(SFOUT(J))
LAMDAI = LAMCAF(SPOT(J),1,1)
TOP = TOPF(SFOUT(J))
PRINP = RHOTPF(SFOUT(J))*AR*TIAP
IF (LTPL.LE.0.1) GO TO 10
RELTOP(J) = PROP(J)/PRINP*(TINP/TOP)**(GAM*EXPO)
GO TO 20
10 RELTOP(J) = 1.-LOSOUT(J)
20 CONTINUE
C--DISTRIBUTE LOSS ON ORTHOGONAL MESH WITHIN BLADES, ONE ORTHOGONAL
C--MESH LINE AT A TIME
C
DO 30 J=1,MHTPI
30 SF(J) = (ROM(MM,J)**2-RCM(MM,1)**2)/(ROM(MM,MHTPI)**2-RCM(MM,1)**2)
CALL SPLINT1(SFOUT,RELTOP,NOUT,SF,MHTPI,PRATIC,PRAC)
DO 40 J=1,MHTPI
ILEJ = ILE(J)
SLENTH = STEOM(J)-SLEM(J)
DC 40 I=ILEJ,MM
DEL = AMIN1(SLENTH,SOM(I,J)-SLEM(J))
LOSS(I,J) = (1.-PRATIO(J))*DEL/SLENTH
40 RTH(I,J) = RTH(I,J) + (1.-LOSS(I,J))
IF (IDEBUG.LE.0) RETURN
WRITE(NWRIT,1010)
WRITE(NWRIT,1000) (I,J,SOM(I,J),SOM(I,J),RTH(I,J),DTOT(I,J),
1PLOSS(I,J),CPI(J,J),SPI(J,J),[I=1,MM],J=1,MHTPI)
RETURN
1000 FORMAT(2I6,7I16,6)
1010 FORMAT(I1,15X,IH15,1HJ,13X,PHI,12X,PHI)
END

SUBROUTINE MEPLT
C
C--MEPLT PLOTS THE BLADE GEOMETRY AND THE GENERATED CRTHGCGNAL MESH
C
COMMON/INPUT/GAM,AR,MSFL,OMEGA,RECFAC,VELTOL,FNEW,ONEW,MBI,MBO,
1 MM,MHT,NBL,NHUR,NTP,NIN,NCUT,NRLP,NPPP,NCSTAT,NSL,LSFR,
2 LTPL,LAMVT,IMESH,ISLINE,ISTATL,IPLOT,ISUPER,ITSON,IDEBUG,
3 ZMIN,ZMAX,7CMIN,7CMAX,ZMIN,ZIN,7ZIN,7ZOUT,7ZOUT,7ZHUR(50),
4 RHU(50),ZHUR(50),RTIP(50),SFOUT(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTIN(50),SFOUT(50),RADOUT(50),PRIP(50),LOSOUT(50),
6 LAMOUT(50),VTOUT(50),ZHST(50),ZST(50),FLFR(50),
7 ZPL(50,50),ZPL(50,50),ZPL(50,50),ZPL(50,50),ZPL(50,50),ZPL(50,50)
COMMON/CALCON/MMM1,MHTPI,CP,EXPO,TGROG,PITCH,CURVHI,CURVTI+
1 CURVHO,CURVTO,RHIN,RTIN,RCUT,RTETO,RLEH,RTET,RTETO
2 ZLE(50),ZLE(50),ZTE(50),ZTE(50),ZLEOM(101),ZLECM(101),
3 SLEM(101),THLEOM(101),THTECM(101),RTOM(101),STEO(101),
140
4 TMTEDM(101), ILE(101), IVE(101), ZOM(100, 101), RCM(100, 101), 5 TOM(100, 101), ATH(100, 101), OTHOS(100, 101), 6 OTN(T100, 101), PLOSO(100, 101), CPHI(100, 101), 7 COMMON/PLC/CDM/RLANG, ZRANG, RRRNG, RRRNG, RRRNG, ZHPLT(100), 8 RHPLT(100), SPLT(100), RPLT(100), RLPLT(100), 9 TPLT(100), 2 RPLT(100) 10 DIMENSION TITL1(15), TITL2(10), TITL3(3), TITL4(3), ZTM(101), 11 RTEN(101) 12 DATA TITL1/'HUB', 'SHR', 'D UD', 'ANC', 'BLA', 'CE B', 'UND', 'ARIE', 13 'S6C1', 'SRBI', 'EN ME', 'R1DI', 'ORNAL', 'PLA', 'NE ', '/ 14 DATA TITL2/'ORTN', 'OGON', 'AL M', 'ESH', 'C1SL', '2IN', 'MERI', 'DIO', 15 'AL P', 'LANE', '/ 16 DATA TITL3/'C', 'IREC', 'TION', '/ 17 DATA TITL4/'R', 'D', 'IREC', 'TION', '/ 18 DATA SYM'*/'/ 19 DATA SYM'*/' IF (IPLT.LE.0) RETURN 20 C--CERTAIN PLOT AUCNARIES, AND SCALE THE PLOT 21 CALL PTDROY 22 C--PLOT PLANE GEOMETRY AND PLOT ORTHOGONAL MESH 23 CALL LRMN(1, 0, 1, 0, 2, 0, 1, 0) 24 CALL LRMN(1, 0, 1, 0, 2, 0, 1, 0) 25 CALL LRAGNC(1, 0, 1, 0, 2, 0, 1, 0) 26 IPLT= 1 27 CALL LRLEGM(TITL1, 60, 0, 1, 3, 0, 7, 0, 0) 28 IF (IPLT.EQ.2) CALL LRLEGM(TITL2, 4, 0, 3, 4, 0, 7, 0, 0) 29 CALL LRCHSZ(2) 30 CALL LRLEGM(TITL3, 12, 0, 4, 5, 1, 5, 0, 0) 31 CALL LRLEGM(TITL4, 12, 1, 0, 4, 5, 0, 0) 32 CALL LRCHSZ(4) 33 CALL LRCURV(ZHPLT, RHPLT, 100, 2, SYM, 0, 0) 34 CALL LRCURV(ZSPLT, RSPLT, 100, 2, SYM, 0, 0) 35 CALL LRCURV(ZLPLT, RLPLT, 100, 2, SYM, 0, 0) 36 CALL LRCURV(ZTPLT, RTPLT, 100, 2, SYM, 0, 0) 37 IF (IPLT.EQ.2) GO TO 20 38 CALL LRCURV(ZHUB, RHUB, NHUB, 4, SYM, 0, 0) 39 CALL LRCURV(ZTIP, RTIP, NTIP, 4, SYM, 0, 0) 40 DO 15 JN=1, NBLPL 41 CALL LRCURV(ZB(1..JN), RB(1..JN), NPPP, 2, SYM, 0, 0) 42 CALL LRCURV(ZLE, RLE, NBLPL, 3, SYM, 0, 0) 43 CALL LRCURV(ZTE, RTE, NBLPL, 3, SYM, 1, 0) 44 IPLT= 2 45 GO TO 10 46 C--PLOT VERTICAL MESH LINES 47 DO 40 I=1, MM 48 ZTM(1)= ZCM(I, 1) 49 RTEN(1)= ROM(I, 1) 50 DO 30 J=2, MHTP1 51 ZTM(J)= ZCM(I, J) 52 30 RTEN(J)= ROM(I, J) 53 40 CALL LRCURV(ZTEM, RTE, MHTP1, 2, SYM, 0, 0) 54 C--PLOT HORIZONTAL MESH LINES 55 ECP= 0, 0 56 DO 50 J=2, MHT 57 IF (J.EQ.MHT) EOP=1, 0 58 CALL LRCURV(ZCM(1, J), ROM(1, J), MM, 2, SYM, EOP) 59 CALL LRCURV(ZTEM, RTE, 0, 1, SYM, 1, 0) 60 RETURN 61 END
SUPROUTINE PTBDRY

C--PTBDRY OBTAINS THE HUB AND SHROUD AND BLADE LEADING AND TRAILING EDGE C--BOUNDARIES FOR PLOTTING, AND SCALES THE PLOT

C

COMMON/INPUTT/GAM,AR,MFL,OMEGA,REC FC,VELTOL,FNEW,DNEW,MAI,MB0,   
1 MM,MHT,INL,NHUR,NTIP,IN,NOTR,NPL,NPP,NSST,NSL,LSFR,   
2 LTPL,LMWT,TMESH,ISLINE,ISTAT,IPLOT,ISUP,ISON,IDEBUG,   
3 TMNI,TOMI,ZCMBO,ZCMOUT,ZHIN,TIN,ZHOUT,TCUT,ZHUR(50),   
4 RHUR(50),ZTIP(50),RTIP(50),SFN(50),RADIN(50),TIP(50),PRIP(50),   
5 LAMIN(50),VTHIN(50),SFOUT(50),RACOUT(50),PROP(50),LOOUT(50),   
6 LAMOUT(50),VTHOUT(50),ZHST(50),ZST(50),FLFR(50),   
7 ZPL(50,50),RLL(50,50),THRL(50,50),TNRL(50,50)   

COMMON/CALCN/MM1,MHTP1,CP,EXPON,TCRG,PITCH,CURVH,CURVT,   
1 CURVH,CURVT,RHIN,RTIN,RHOUT,RTOUT,RLEH,RTET,   
2 ZLE(50),RLE(50),ZTE(50),RTE(50),ZLEOM(101),RLEOM(101),   
3 ZTEOM(101),THLEOM(101),ZTEC(101),RTEC(101),STEM(101),   
4 THTEOM(101),TLE(101),TTEC(101),ZTOM(100,101),RM(100,101),   
5 SOM(100,101),THOM(100,101),BTH(100,101),DTHDS(100,101),   
6 DTHERT(100,101),PLSST(100,101),CPI(100,101),SPI(100,101)   

COMMON/PLTCOM/ZLNGZ,RZZNG,RRNG,RTNG,ZHPLT(100),RHTL(100),   
1 ZSPLT(100),RSPLT(100),ZLPLT(100),RLPLT(100),ZTPLT(100),   
2 RTPLT(100)   

DIMENSION AAA(100)

C--OBTAIN PLOT POINTS ON HUB

C

DELZ = (ZHUB(NHUR)-ZHUB(1))/99.   
ZHPLT(1) = ZHUB(1)   
DO 10 I=2,100   
10 ZHPLT(I) = ZHPLT(I-1) + DELZ   
CALL SPLINT(ZHUB,RHUB,NHUR,ZHPLT,100,RHPLT,AAA)

C--OBTAIN PLOT POINTS ON SHROUD

C

DELZ = (ZTIP(NTIP)-ZTIP(1))/99.   
ZSPLT(1) = ZTIP(1)   
DO 20 I=2,100   
20 ZSPLT(I) = ZSPLT(I-1) + DELZ   
CALL SPLINT(ZTIP,RTIP,NTIP,ZSPLT,100,RSPLT,AAA)

C--OBTAIN PLOT POINTS ON BLADE LEADING EDGE

C

DELZ = (RLEH-RLEH)/99.   
RLPLT(1) = RLEH   
RLPLT(100) = RLEH   
DO 30 J=2,99   
30 RLPLT(J) = RLPLT(J-1) + DELZ   
CALL SPLINT(RLEH,ZLE,NRLPL,RLPLT,100,ZLPLT,AAA)

C--OBTAIN PLOT POINTS ON BLADE TRAILING EDGE

C

DELZ = (RTET-RTEH)/99.   
RTPLT(1) = RTEH   
RTPLT(100) = RTET   
DO 40 J=2,99   
40 RTPLT(J) = RTPLT(J-1) + DELZ   
CALL SPLINT(RTEH,ZTE,NRLPL,RTPLT,100,ZTPLT,AAA)

C--CALCULATE THE RANGE OF THE PLOT

C

142
SUBROUTINE VRDRY(LCC,TIPF,RH0IF,LMCAF)

C-- VRDRY CALCULATES THE DISTRIBUTION OF STREAM FUNCTION ALONG THE
C-- UPSTREAM AND DOWNSTREAM BOUNDARIES OF THE ORTHOGONAL MESH
C
COMMON SW1,SW2,ITER,ENC,NREAC,NWRT
COMMON/INPUT/RELAM,AR,MSFL,OMEGA,REDFA,C,VELTCL,FNEW,DNEW,MBI,MBO,
  MM,MHT,HRR,NHR,NTIP,NIN,NOUT,NRLPL,NPPPL,NOSTNL,LSFR,
  LTPI,LAMUT,IMESH,IMLINE,ISTATL,IPLOT,I SupER,ITSON,IIDBUG,
  ZM,MBI,ZOMB,7CM,7TC,7HT,7THIN,7HTOUT,7TCUT,7WH(50),
  RHUR(50),RTIP(50),SFN(50),RADIN(50),TIP(50),PROP(50),
  LAMIN(50),VTHIN(50),SFOUT(50),RACOUT(50),LOSOUT(50),
  LAMOUT(50),VTHOUT(50),ZHT(50),ZHTOUT(50),FLFR(50),
  ZR(50,50),NR(50,50),THBL(50,50),TNBL(50,50),
 COMMON/CALCCM/MM1,MHTPL,CP,EXCPN,TGROG,PITCH,CURV,ICURV1,
  CURVH1,CURVTO,HIN,HOUT,RHEL,RLET,REP,RTET,
  TLE(50),RLE(50),ZTE(50),RTE(50),ZLEDM1(101),RLEDM1(101),
  SLEDOM1(101),THLEDM1(101),ZTEOM1(101),RTEOM1(101),
  THTOM1(101),ITEL(101),ITE(101),ZOM(100,101),RCH(100,101),
  SM(100,101),TCM(100,101),PRH1(100,101),DPHT(100,101),
  DTHDT(100,101),PLLOSS(100,101),CPHI(100,101),SFIN(100,101),
 COMMON/VARCOM/4,100,101),UOM(100,101),K1(100,101),RH1(100,101),
  WSURS(100,101),WSSUR(100,101),WSURF(100,101),WSURF(100,101),
  WSURS(100,101),WSSUR(100,101),WHT(100,101),VHT(100,101),W(100,101),
  ALP(100,101),RAYA(100,101),WWC(100,101),CURV(100,101),
  WLSURF(100,101),WTSURF(100,101),CAMP(100,101),SAMP(100,101),

ZLRNG = AMIN1(ZHUR(1),RTIP(1))
ZRRNG = AMAX1(ZHUR(NHR),RTIP(NTIP))
DO 48 J=1,MHTPL
ZLRNG = AMIN1(ZLRNG,ZOM(1,J))
48 ZRRNG = AMAX1(ZRRNG,ZOM(MMM,J))
DELZ = ZRRNG-ZLRNG
ZLRNG = ZLRNG-0.05*DELZ
ZRRNG = ZRRNG+0.05*DELZ
RPRNG = RHUR(I)
DO 50 I=2,NHR
RPRNG = AMIN1(RPRNG,RHUR(I))
RTIP = RTIP(I)
DO 60 I=2,NTIP
RTRNG = AMAX1(RTRNG,RTIP(I))
DELR = RTRNG-RPRNG
RPRNG = RPRNG-0.05*DELR
RTRNG = RTRNG+0.05*DELR
C
C-- CHOOSE MAXIMUM RANGE, AND EXPAND RANGE IN THE OTHER DIRECTION
C
DM02 = 1.1*ABS(DELZ-DELR)/2.
IF (DELR.GT.DELZ) GO TO 70
RTRNG = RTRNG+DM02
RPRNG = RPRNG-DM02
RETURN
70 ZLRNG = ZRRNG+DM02
ZLRNG = ZLRNG-DM02
RETURN
END

C
COMMON SAW,SRE,ITER,ENC,NREAC,NWRT
COMMON/INPUT/RELAM,AR,MSFL,OMEGA,REDFA,C,VELTCL,FNEW,DNEW,MBI,MBO,
  MM,MHT,HRR,NHR,NTIP,NIN,NOUT,NRLPL,NPPPL,NOSTNL,LSFR,
  LTPI,LAMUT,IMESH,IMLINE,ISTATL,IPLOT,I SupER,ITSON,IIDBUG,
  ZM,MBI,ZOMB,7CM,7TC,7HT,7THIN,7HTOUT,7TCUT,7WH(50),
  RHUR(50),RTIP(50),SFN(50),RADIN(50),TIP(50),PROP(50),
  LAMIN(50),VTHIN(50),SFOUT(50),RACOUT(50),LOSOUT(50),
  LAMOUT(50),VTHOUT(50),ZHT(50),ZHTOUT(50),FLFR(50),
  ZR(50,50),NR(50,50),THBL(50,50),TNBL(50,50),
 COMMON/CALCCM/MM1,MHTPL,CP,EXCPN,TGROG,PITCH,CURV,ICURV1,
  CURVH1,CURVTO,HIN,HOUT,RHEL,RLET,REP,RTET,
  TLE(50),RLE(50),ZTE(50),RTE(50),ZLEDM1(101),RLEDM1(101),
  SLEDOM1(101),THLEDM1(101),ZTEOM1(101),RTEOM1(101),
  THTOM1(101),ITEL(101),ITE(101),ZOM(100,101),RCH(100,101),
  SM(100,101),TCM(100,101),PRH1(100,101),DPHT(100,101),
  DTHDT(100,101),PLLOSS(100,101),CPHI(100,101),SFIN(100,101),
 COMMON/VARCOM/4,100,101),UOM(100,101),K1(100,101),RH1(100,101),
  WSURS(100,101),WSSUR(100,101),WSURF(100,101),WSURF(100,101),
  WSURS(100,101),WSSUR(100,101),WHT(100,101),VHT(100,101),W(100,101),
  ALP(100,101),RAYA(100,101),WWC(100,101),CURV(100,101),
  WLSURF(100,101),WTSURF(100,101),CAMP(100,101),SAMP(100,101),

ZLRNG = AMIN1(ZHUR(1),RTIP(1))
ZRRNG = AMAX1(ZHUR(NHR),RTIP(NTIP))
DO 48 J=1,MHTPL
ZLRNG = AMIN1(ZLRNG,ZOM(1,J))
48 ZRRNG = AMAX1(ZRRNG,ZOM(MMM,J))
DELZ = ZRRNG-ZLRNG
ZLRNG = ZLRNG-0.05*DELZ
ZRRNG = ZRRNG+0.05*DELZ
RPRNG = RHUB(I)
DO 50 I=2,NHR
RPRNG = AMIN1(RPRNG,RHUR(I))
RTIP = RTIP(I)
DO 60 I=2,NTIP
RTRNG = AMAX1(RTRNG,RTIP(I))
DELR = RTRNG-RPRNG
RPRNG = RPRNG-0.05*DELR
RTRNG = RTRNG+0.05*DELR
C
C-- CHOOSE MAXIMUM RANGE, AND EXPAND RANGE IN THE OTHER DIRECTION
C
DM02 = 1.1*ABS(DELZ-DELR)/2.
IF (DELR.GT.DELZ) GO TO 70
RTRNG = RTRNG+DM02
RPRNG = RPRNG-DM02
RETURN
70 ZLRNG = ZRRNG+DM02
ZLRNG = ZLRNG-DM02
RETURN
END
5 RHOAV(100,101), DELRHOF(100,1C1), FR(100,101), CFCM(100,101),
6 XIOM(100,101), ZETOM(100,101), PLDU(100,101),
DIMENSION TIPROY(101), RHOIP(101), LAMCA(101), AA(101), PB(101),
7 CCA(101), CCR(1C1), WDRY(1C1), RDY(4),
REAL LAMDA, LAMCAF, MSFL
LOGICAL REPEAT
EXTERNAL TIPF, RHOIPF, LAMDAF
DATA RDY/4*HINLE, 4HT, 4HOUTL, 4MET /
C--SET INITIAL WHUB AND DELMAX
C
IL= 1
IF (LOC.EQ.1) IL=3
CURVH = CURV+1
CURVT = CURVTI
IF (LOC.EQ.1) GO TO 10
CURVH = CURVHO
CURVT = CURVT0
10 J= 1
RMEAN = (ROM(LOC,1)+ROM(LOC,MHTPI))/2.
RHOIP(1)= RHOIPF*(5)
WSM= MSFL/RHOIP(1)/RMEAN/PITCH/TOM(LOC, MHTPI)
WTHETA= LAMCAF(5,LOC,1)/RMEAN-OMEGA*RMEAN
WHUR = SORT(WSM**2+WTHETA**2)
DELMAX = WHUR/20.
RTCLER = 1.E-4
C--CALCULATE INITIAL ESTIMATE OF TIP, RHOIP, AND LAMDA
C
RH2 = ROM(LOC,1)**2
DELR2 = ROM(LOC,MHTPI)**2-RH2
DC 20 J=1,MHTPI
WFLF = (ROM(LOC,J)**2-RH2)/DELR2
UCM(LOC,J) = WFLF
TIPBDY(J)= TIPF(WFLF)
RHOIP(J)= RHOIPF(WFLF)
20 LAMDA(J)= LAMCAF(WFLF,LOC,J)
NCOUNT = 0
C--CALCULATE COEFFICIENTS A, B, AND C FOR THE VELOCITY GRADIENT EQUATION
C
30 DO 40 J=1,MHTPI
AA(J) = CURVH*TOM(LOC,J)/TOM(LOC,MHTPI)*(CURVT-CURVH)
OMR2 = OMEGA*ROM(LOC,J)**2
BB(J) = -(LAMDA(J)-OMR2)/ROM(LOC,J)**2*(AA(J)*(LAMDA(J)-OMR2)+
1 *(LAMDA(J)+OMR2)/ROM(LOC,J)*CPHI(LOC,J))
40 CONTINUE
DO 50 J=1,MHTPI
CC(J)= CP*(TIPBDY(J+1)-TIPBDY(J))-OMEGA*(LAMDA(J+1)-LAMDA(J))
50 CC(J)= (CP-AR)*(TIPBDY(J+1)-TIPBDY(J)+APR/(RHOIP(J)+RHOIP(J+1)))*
1*(TIPBDY(J)+TIPBDY(J+1))*(RHOIP(J+1)-RHOIP(J))
C--SOLVE THE VELOCITY GRADIENT EQUATION ALONG THE BCUNARY
C
REPEAT = .FALSE.
UCM(LOC,1) = 0.
60 INT= 1
70 WPCRY(1) = WHUR
NCOUNT = NCOUNT+1
程序代码，具体的代码行如下：

```plaintext
! WSO = WDRY(J)**2
TWLMR = 2.*OMEGA*LAMBDA(J) - (OMEGA*ROM(LOC, J))**2
TTIP = 1. - (WSO + TWLMR)/CP/TIPBDY(J+1)/2.
IF (TTIP < 0.) GO TO 100
RHOW = RHOIP(J+1)TTIP**EXPN*WDRY(J)
SINRTA = (LAMBDA(J)/ROM(LOC, J) - OMEGA*ROM(LOC, J))**2
IF (ABS(SINRTA) > 1.) GO TO 90
CCRTA = SCRT(1. - SINRTA**2)
RVA = RHOW*COSRTA*ROM(LOC, J)*PITCH
DC 80 J=1,MHT
DELTA = TDC(LOC, J) - TCM(LOC, J)
CC = CCAL(J) - CCBT(J) + TTIP
WAS = WADRY(J)*1. + AA(J)*DELTA + BB(J) + WDRY(J) + DELTA*CC/WDRY(J)
WSO = WAS**2
TWLMR = 2.*OMEGA*LAMBDA(J+1) - (OMEGA*ROM(LOC, J+1))**2
TTIP = 1. - (WSO + TWLMR)/CP/TIPBDY(J+1)/2.
CC = CCAL(J+1) - CCBT(J) + TTIP
WASS = WADRY(J+1)*AA(J+1)*DELTA + BB(J+1) + WAS + DELTA*CC/WAS
WDRY(J+1) = (WAS + WASS)/2.
WSO = WDRY(J+1)**2
TTIP = 1. - (WSO + TWLMR)/CP/TIPBDY(J+1)/2.
IF (TTIP < 0.) GO TO 100
RHOW = RHOIP(J+1)TTIP**EXPN*WDRY(J+1)
SINRTA = (LAMBDA(J+1)/ROM(LOC, J+1) - OMEGA*ROM(LOC, J+1))**2
IF (ABS(SINRTA) > 1.) GO TO 90
CCRTA = SCRT(1. - SINRTA**2)
RVAS = RHOW*COSRTA*ROM(LOC, J+1)*PITCH
UOM(LOC, J+1) = (RVA + RVAS)*DELTA/2. + UCM(LOC, J)
RO RVA = RVAS

C-- CHECK CONTINUITY AND ESTIMATE NEW VALUE FOR W AT MHT
IF (IND.GE.6.AND.ABS(MSFL - UOMCLOCMHTP1)).LE.MSFL**RTOLER) GO TO 120
CALL CONTIN(WHISB, UCM(LOC, MHTP1), IND, Z, MSFL, DELMAX)
IF (IND.EQ.10) GO TO 120
GO TO 110
90 WHISB = WHISB + 0.5*DELMAX
IF (NCOUNT.LT.1000) GO TO 60
GO TO 110
100 WHISB = WHISB - 0.5*DELMAX
IF (NCOUNT.LT.1000) GO TO 60
110 WRITE(NWRT, 1010) RDY(1), ADY(1+I)
STOP

C-- SOLUTION OBTAINED. UPDATE TIP, RHOIP, AND LAMBDA
DO 130 J=1,MHT
FRAC = UOM(LOC, J)/MSFL
UCM(LOC, J) = FRAC
TVAR = TIPF(FRAC)
IF (ABS(TVAR - TIPBDY(J)).GE.TIPBDY(J)) REPEAT = TRUE.
TIPBDY(J) = TVAR
TVAR = RHOIPF(FRAC)
IF (ABS(TVAR - RHOIP(J)).GE.RHOIP(J)) REPEAT = TRUE.
RHOIP(J) = TVAR
TVAR = LAMDAF(FRAC, LOC, J)
IF (ABS(TVAR - LAMBDA(J)).GE.ABS(LAMBDA(J))) REPEAT = TRUE.
```

145
130 LAMPA(I,J) = TVAR
WHAR = WRDYM(I)
IF (REPEAT.AND.NCOUNT.GE.1000) GO TO 110
IF (REPEAT) GO TO 30
IF (IND.NE.10) RETURN
WRTFE(NRTW,1000) BOV(IL),BOV(IL+1),UOM(LOC,MHTP1)
STOP
1000 FORMAT (26HL PASSAGE IS CHOKEC AT THE,2A4,21H WITH A MASS FLOW OF
14.6)
1010 FORMAT (2HL,2A4,39H BOUNDARY CONDITIONS CANNOT BE OBTAINED)
END

SUBROUTINE INIT

C INIT ASSIGNS INITIAL VALUES TO THE ARRAY VARIABLES

C
COMMON/INPUTT/GAM,AR,MSFL,OMEC,RECFA,VEHTOL,FNEW,DNEW,MHI,MBO,
  MM,MHT,RLR,HUB,NTP,NIN,NCUT,NPLP,NPPP,NOSTAT,NST,LSFR,
  LTP,LAMVT,IMESH,ISLINE,ISTATL,IPLOT,ISUPER,ITSON,IDEBUG,
  3?MIN,?OMB,?OMDOUT,?MIN,?MDIN,?MDOUT,?MDIN,?MDOUT,
  4RUB(50),ZTIP(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
  5LAMIN(50),VTHIN(50),SOUT(50),RACOUT(50),PROP(50),LOSOUT(50),
  6LAMOUT(50),VTHOUT(50),ZHST(50),ZTST(50),FLFR(50),
  7ZBL(50,50),ZRL(50,50),ZTH(50,50),ZTST(50,50),
COMMON/CALCCN/MMM1,MHTPL,CP,EXFON,TGROG,PMICH,CURVHI,CURVTI,
  1CURVH0,CURVTH,RLHRTIN,RLHRTOUT,RLHRTIN,RLHRTOUT,
  2ZLE(50),ZLE(50),ZTE(50),ZTE(50),ZLEOM(101),ZLEOM(101),
  3SLEOM(101),SLEOM(101),ZTEOM(101),ZTEOM(101),
  4THTEOM(101),THTEOM(101),THE(101),THE(101),
  5SCH(100,101),SCH(100,101),SCH(100,101),SCH(100,101),
  6SCH(100,101),SCH(100,101),SCH(100,101),SCH(100,101),
COMMON/VARCOM/A(4,100,101),UOM(100,101),K(100,101),RHO(100,101),
  1WSURF(100,101),WSURF(100,101),WSURF(100,101),WSURF(100,101),
  2WSURM(100,101),WSURM(100,101),WSURM(100,101),WSURM(100,101),
  3ALPA(100,101),ALPA(100,101),ALPA(100,101),ALPA(100,101),
  4LAMDA(100,101),LAMDA(100,101),LAMDA(100,101),LAMDA(100,101),
  5XOM(100,101),XOM(100,101),XOM(100,101),XOM(100,101),
  6XOM(100,101),XOM(100,101),XOM(100,101),XOM(100,101),
REAL K
RHOIPF = RHOIPF(0.)
DO 10 J=1,MMH
DO 10 I=1,NN
WSUR(I,J) = CP'H(I,J)
WSUR(I,J) = SPHI(I,J)
WSUR(I,J) = 1.
W(I,J) = 0.
WTH(I,J) = 0.
VTH(I,J) = 0.
RHOF(I,J) = RHOF
RHOF(I,J) = RHOF
DELPHO(I,J) = DELPHO
DELPHO(I,J) = DELPHO
10 CONTINUE
DIMENSION OVTVD(100,101)

REAL MSFL,K,KNEW

C--CALCULATE COEFFICIENTS AND CONSTANTS FOR FINITE DIFFERENCE EQUATIONS

WRITE(NWRIT,1030) ITER

DCPHA = 0.

DCHX = -1.E20

DMIN = 1.E20

DO 30 J=2,MM1

H4 = SOM(2,J)-SCM(1,J)

DO 30 I=2,MM1

H1 = TOM(I,J)-TOM(I,J-1)

H2 = TOM(I,J+1)-TOM(I,J)

H3 = H4

H4 = SCM(I+1,J)-SCM(I,J)

AO = 2.*H1/H2+2.*H3/H4

C1 = H1+H2

C2 = H3+H4

D1 = (RTH(I,J+1)-RTH(I,J-1))/ETH(I,J+1)+(RHO(I,J+1)-RHO(I,J-1))/IRHO(I,J)

D1 = D1*C1+CPHI(I,J)+ROM(I,J)*(SPHI(I+1,J)-SPHI(I-1,J))/C2

ICPHI(I,J)

D2 = (RTH(I+1,J)-RTH(I-1,J))/ETH(I,J)+(RHO(I+1,J)-RHO(I-1,J))/ICPHI(I,J)

RETURN

END

SUBROUTINE COEF

C--COEF CALCULATES COEFFICIENTS A AND K,

C--FOR THE SYSTEM OF MATRIX EQUATIONS, A*U=K

COMMON SRW,SRE,ITER,IEND,NREAD,NWRITE

COMMON/INPUT/GAM,AR,MSFL,OMEGA,RECFA,VELTOL,FNEW,DNEW,MBI,MBO,1
MM,MMT,NBL,NHR,NTIP,NIN,NCT,NPL,NPPP,NCSTAT,NSL,LSL,LSR,
2
LTPL,LMVT,IMESH,ISLTAL,IPLOT,ISUPER,ITSON,INDERUG,
3
ZMIN,ZMBI,ZCMR,ZOUT,ZHIN,THIN,HMIN,HMIN,HOUT,THOUT,
4
RHUM(50),ZP(50),RT(50),SF(50),RADIN(50),TIP(50),PRP(50),
5
LAMIN(50),VTIN(50),SFOUT(50),RAOUT(50),PROP(50),LOSOUT(50),
6
LAMOUT(50),VTOUT(50),ZHST(50),TST(50),FLFR(50),
7
R0(50,50),RAL(50,50),THBL(50,50),TNBL(50,50)

COMMON/CONC/MM1,MTPI,CP,EXPON,TGROG,PITCH,CURVH,CURVTI,
1
1

CURVHO,CURVTO,RHIN,RTIN,RTOUT,RYSH,RYST,RTET,
2
ZLE(50),ZHE(50),ZTE(50),ZTE(50),ZLEOM(101),ZLEOM(101),
3
SLEOM(101),THLEOM(101),ZTEC(101),RTLEM(101),STEOM(101),
4
THTEOM(101),ILE(101),ITET(101),ZOM(100,101),RZM(100,101),
5
SCM(100,101),TM(100,101),RTH(100,101),DTKH(100,101),
6
OTH(100,101),GLOS(100,101),CPUAT(100,101),SPH(100,101),

COMMON/VARCOM/A(4,100,101),UOM(100,101),K(100,101),RHO(100,101),
1
WSUB(100,101),WSUBT(100,101),WSUBR(100,101),WSUBR(100,101),
2
WSUB(100,101),WTH(100,101),WTH(100,101),WTH(100,101),
3
ALPHA(100,101),BETA(100,101),WMCRC(100,101),CURV(100,101),
4
WSURF(100,101),WTSURF(100,101),CAMP(100,101),SAMP(100,101),
5
RHOAV(100,101),DLR(100,101),FRI(100,101),DFCM(100,101),
6
ITIM(100,101),ZETATM(100,101),DLT(100,101)

DIMENSION OVTVD(100,101)

REAL MSFL,K,KNEW

C--CALCULATE COEFFICIENTS AND CONSTANTS FOR FINITE DIFFERENCE EQUATIONS

WRITE(NWRIT,1030) ITER

DCPHA = 0.

DCHX = -1.E20

DMIN = 1.E20

DO 30 J=2,MM1

H4 = SOM(2,J)-SCM(1,J)

DO 30 I=2,MM1

H1 = TOM(I,J)-TOM(I,J-1)

H2 = TOM(I,J+1)-TOM(I,J)

H3 = H4

H4 = SCM(I+1,J)-SCM(I,J)

AO = 2.*H1/H2+2.*H3/H4

C1 = H1+H2

C2 = H3+H4

D1 = (RTH(I,J+1)-RTH(I,J-1))/ETH(I,J)+RHO(I,J+1)-RHO(I,J-1))/IRHO(I,J)

D1 = D1*C1+CPHI(I,J)+ROM(I,J)*(SPHI(I+1,J)-SPHI(I-1,J))/C2

ICPHI(I,J)

D2 = (RTH(I+1,J)-RTH(I-1,J))/ETH(I,J)+RHO(I+1,J)-RHO(I-1,J))/
1RH(I,J)
D2 = D2/C2+SPHI(I,J)/RCM(I,J)-(SPHI(I,J+1)-SPHI(I,J-1))/C1/
CPHI(I,J)
KNEW = XIOM(I,J)*W(I,J)*2+ZETOM(I,J)
IF(IJE.LT.1)AND(IJE.LT.ITE(J)) GO TO 10
KNEW = KNEW+WTH(I,J)/DSFL*RTH(I,J)*RHC(I,J)*WSUB(I,J)*DLU(I,J)
GO TO 20
10 DVVRT = (ROM(I+1,J)*VTH(1+J,J)-ROM(I,J)*VTH(1,J))/C2*SPHI(I,J)
1+ROM(I,J+1)*VTH(I+1,J)-ROM(I,J-1)*VTH(I,J-1))/C1*CPHI(I,J)
DCHANG = AMAX1(DCANG,ABS(DVVRT-DVVRT(I,J)))
DMAX = AMAX1(DMAX,DVVRT)
DMIN = AMIN1(DMIN,DVVRT)
DVTHDR(I,J) = DNEW*DVRT+(1.-DNEW)*DVTHDR(I,J)
KNEW = KNEW+WTH(I,J)/ROM(I,J)*DVTHDR(I,J)+FR(I,J)
20 KNEW = KNEW*ROM(I,J)/AO*8TH(I,J)/MSFL*RHC(I,J)/WSUB(I,J)
K[I,J] = KNEW
A(I+1,J) = (2./H1+01)/AO/C1
A(I+2,J) = (2./H2+01)/AO/C1
A(3,I+1,J) = (2./H3+02)/AO/C2
A(4,I+1,J) = (2./H4+02)/AO/C2
30 CONTINUE
WRITE(NWRT,1020) DMAX,DMIN,DCHANG
IF (ICERI<0.E+1 RETURN
IF (((ITER/IOEBUG)*ICERUG.NE.I) RETURN
WRITE(NWRT,1010)
DC 40 J=2,MHT
DC 40 I=2,MM1
40 WRITE(NWRT,1000) I,J,(A[I,J],J),I,J=14,K(I,J)
RETURN
1000 FORMAT (216,5G16.6)
1010 FORMAT (1H1//24X,57HCoefficients OF Matrix Equations for Str
TEAM Function/5X,1HI,5X,H1J,6X,4HA(1),12X,4HA(2),12X,4HA(3),12X,
24HA(4),13X,1HK)
1020 FORMAT (///5X,37HMaxIMUM CALCULATED VALUE OF DVTHDR =,G13.5/5X,
137HMinIMUM CALCULATED VALUE OF DVTHDR =,G13.5/5X,37HMaxIMUM CALCU
LATED CHANGE IN DVTHDR =,G13.5)
1030 FORMAT (/////14H Iteration NO.,13,2H :)
END

SURROUTINE SOR
C--SOR SOLVES THE SET OF MATRIX EQUATIONS, A*U=K
C--BY THE SUCCESSIVE OVERRELAXATION TECHNIQUE
C
COMMON SOR,SRE,ITER,ENEC,NERAC,NWRT
COMMON/INPUT/AM,AR,MSFL,OMEGA,REDPAC,VELTCL,FNEW,DNEW,MBI,MBD,
1 MM,MHT,MRN,NNR,NTP,NN,NOUT,NBLPL,NPPP,NSTAT,NSL,LSFR,
2 LTPL,LMW,TMESH,ISLINE,ISATL,PLCIT,ISUPER,ITSON,ICEBUG,
3 I2MNN,12MBI,12MBD,ZSOUT,ZHIN,ZTIN,ZTHOUT,ZTCUT,ZTOUT(50),
4 RHUR(50),RTIP(50),RTIP(50),SFOUT(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTHIN(50),SFOUT(50),RADOUT(50),PRCIP(50),LOSOUT(50),
6 LAMOUT(50),VTHOUT(50),ZHST(50),ZTST(50),FLFST(50),
7 ZAL(50,50),RBL(50,50),TBL(50,50),TNBL(50,50),
COMMON/CALCON/MM1,MHTP1,CP,EXPON,TRGCR,PITCH,CURVHI,CURVTI,
1 CURVHD,CURVTO,PHIN,RTIN,RTOUT,RLIN,RLT,RTEN,RTET,
C--STORE U BOUNDARY VALUES AND SET BOUNDARY VALUE TO ZERO

C--TO CALCULATE OPTIMUM ORF

DO 10 I=2,MMM1
   UCM(I,1) = 0.
10   UOM(I,MHTP1) = 0.

DO 20 J=2,MHT
   UVERT(J,1) = UCM(1,J)
   UOM(I,J) = 0.
   UVERT(J,2) = UOM(MM,J)
   UCM(MM,J) = 0.
20   I=2,MMM1

C--CALCULATE OPTIMUM ORF

DO 30 J=2,MHT
   UOM(J,MHTP1) = 0.
   UOM(MJ,J) = 0.
30   I=2,MMM1

C--RESTORE U BOUNDARY VALUES

DO 50 J=2,MHT
50   UOM(J,MHTP1) = 1.

C--SOLVE MATRIX EQUATION BY SOR

ERROR = 0.

C--REAL K,LMAX,LMIN
IF (ITER.GT.1) GO TO 70

C

DIMENSION UVERT(101,2)

REAL K,LMAX,LMIN

C--BOUNDARY VALUES AND SET BOUNDARY VALUE TO ZERO

C--TO CALCULATE OPTIMUM ORF

DO 10 I=2,MMM1
   UCM(I,1) = 0.
10   UOM(I,MHTP1) = 0.

DO 20 J=2,MHT
   UVERT(J,1) = UCM(1,J)
   UOM(I,J) = 0.
   UVERT(J,2) = UOM(MM,J)
   UCM(MM,J) = 0.
20   I=2,MMM1

C--CALCULATE OPTIMUM ORF

DO 30 J=2,MHT
   UOM(J,MHTP1) = 0.
   UOM(MJ,J) = 0.
30   I=2,MMM1

C--RESTORE U BOUNDARY VALUES

DO 50 J=2,MHT
50   UOM(J,MHTP1) = 1.

C--SOLVE MATRIX EQUATION BY SOR

ERROR = 0.

DO 80 J=2,MHT
80   I=2,MMM1

CHANGE = ORF*(A(1,I,J)*UCM(I,J-1)+A(2,I,J)*UCM(I,J+1)+A(3,I,J)*UCM(I-1,J)
  +A(4,I,J)*UCM(I+1,J))

RATIC = UNEW/UCM(I,J)

LMAX = AMAX1(LMAX,RATIO)

LMIN = AMIN1(LMIN,RATIO)

IF ((LMAX-GT.1.) .LT. LMAX) LMAX=1.

ORFMAX = 2./(1.+SORT(1.-LMAX))

ORFMN = 2./(1.+SORT(1.-LMIN))

C--WRITE (NWRITE,1000) ORF

C--REAL K,LMAX,LMIN
IF (ITER.GT.1) GO TO 70

C

DIMENSION UVERT(101,2)

REAL K,LMAX,LMIN

C--BOUNDARY VALUES AND SET BOUNDARY VALUE TO ZERO

C--TO CALCULATE OPTIMUM ORF

DO 10 I=2,MMM1
   UCM(I,1) = 0.
10   UOM(I,MHTP1) = 0.

DO 20 J=2,MHT
   UVERT(J,1) = UCM(1,J)
   UOM(I,J) = 0.
   UVERT(J,2) = UOM(MM,J)
   UCM(MM,J) = 0.
20   I=2,MMM1

C--CALCULATE OPTIMUM ORF

DO 30 J=2,MHT
   UOM(J,MHTP1) = 0.
   UOM(MJ,J) = 0.
30   I=2,MMM1

C--RESTORE U BOUNDARY VALUES

DO 50 J=2,MHT
50   UOM(J,MHTP1) = 1.

C--SOLVE MATRIX EQUATION BY SOR

ERROR = 0.

DO 80 J=2,MHT
80   I=2,MMM1

CHANGE = ORF*(A(1,I,J)*UCM(I,J-1)+A(2,I,J)*UCM(I,J+1)+A(3,I,J)

149
*I* *O* *m* *I* + *A*(*I* + *J*) *I* *O* *m* *I* + *K*(*I* + *J*) *I* *O* *m* *I* + *k* *O* *m* *I* = *A*(*I* + *J*) *I* *O* *m* *I* + *CHANGE

1 IF *ERROR* GT 1.0E-5 GO TO 70
2 RETURN
3
4 1000 FORMAT (1//5X,40H CALCULATED OVERRELAXATION FACTOR (ORF) = , *F*7.3)
5 END

SURROUTINE NEWRHO

C

C -- NEWRHO CALCULATES VELOCITY COMPONENTS, VELOCITY MAGNITUDE,
C -- AND NEW DENSITY AT EACH MESH POINT

C

COMMON SRW, SRE, ITER, IEND, NREA, NWRIT
COMMON INPUT/GAM, AR, MSFL, BMEGA, REDFAC, VELCL, FN EW, DNEW, MBI, MBD,
1 MM, MHT, NLN, NBUT, NTIP, NIN, NOUT, NLRL, NPPP, NOSTAT, NSL, LSFR,
2 LTP, LAVMT, IMSH, ISLINE, ISTATL, IPLICIT, ISUPER, ITSON, IDERUG,
3 ZMIN, ZOMBT, ZCMO, ZOMOUT, ZIN, ZTIN, ZHOUT, ZTCUT, ZHUB(50),
4 RHMIN(50), ZTIP(50), RTIP(50), SFIN(50), RADIQ(50), TIP(50), PRIP(50),
5 LAMIN(50), VTWIN(N), SFOUT(50), RADOUT(50), PRP(50), LOUT(50),
6 LAMOUT(50), VTHOUT(N), ZHTST(50), ZTST(50), FLFR(50),
7 ZBL(50, 50), ZBL(N), THBL(50, 50), TNBL(50, 50)
COMMON/CA CTON/MMM1, MMTH, P, CP, EXPON, TGROG, PITCH, CURVHI, CURVTI,
1 CURVHO, CURVTO, RHI, RTHA, RTIPF, PROPF, RHOCPF

C

C -- REINITIALIZE LAMDAF, RVTHTA, TIPF, RHF, CP, RHOCPF

C

1 IF (LAMVT .EQ. 0.0) GO TO 10
2 CALL LAMVT
3 CALL RVTHTA
4 CALL TIPF
5 CALL RHOF
6 CALL RHONT

C

C -- CALCULATE PARTIAL DERIVATIVES OF RCTHALPY (ROT1), AND RELATIVE TOTAL

150
C--PRESSURE (PREL) IN THE S DIRECTION

10 DO 30 I=1,MPHT1
20 DO 10 I=1,MM
   TPTPTIP(I) = 1.-(2.*OMEGA*LAMDAF(UOM(I,J),I,J)-(OMEGA*ROM(I,J))**2)
1/2./CP/TIPF(UOM(I,J))
   IF (TPTPTIP(I).LT.C.) GO TO 80
   PREL(I) = RHOPF(UOM(I,J))*AR*TIPF(UOM(I,J))**2*(GAM*EXPN(I)**(I)**12.)*
SLOSS(I,J)
20 RTI(I) = CP*TIPF(UOM(I,J))-(OMEGA*LAMDAF(UOM(I,J),I,J)-(OMEGA*ROM(I,J))**2)/
1/2./CP/TIPF(UOM(I,J))
   IF (TPTPTIP(I),LT.C.) GO TO 80
   PREL(I) = RHOPF(UOM(I,J))*AR*TIPF(UOM(I,J))**2*(GAM*EXPN(I)**(I)**12.)*
SLOSS(I,J)
30 CALL SLOPES(SCM(I,J),RTI,MM,ECS(I,J))
C--CALCULATE WSURF FROM THE PARTIAL CF UOM WITH RESPECT TO S USING THE C--AVERAGE BLADE-TO-BLADE DENSITY FCR CONTINUITY

DC 40 J=1,MPHT1
   CALL SPLINE(SCM(I,J),UOM(I,J),CP,DUDS,AAA)
40 DO I=1,MM
   WSURF(I,J) = (DUDS(I)/ROM(I,J)*TH(I,J))*MSFL-CFEM(I,J)*
   DELRHO(I,J)/12.*COS(RETA(I,J))*SAMP(I,J)/(RHCAV(I,J))
40 CONTINUE
C--CALCULATE DERIVATIVES IN THE T DIRECTION OF THE SAME VARIABLES, AND C--CALCULATE NEW VELOCITIES AND NEW DENSITY

DC 60 I=1,MM
    CALL SLPES(TVERT,PREL,MPHT1,CPDT)
    CALL SLOPES(TVERT,PREL,MPHT1,DPDT)
60 DO J=1,MPHT1
    WSURF(I,J) = (DUDS(I)/ROM(I,J)*TH(I,J))*MSFL-CFEM(I,J)*
    DELRHO(I,J)/12.*COS(RETA(I,J))*SAMP(I,J)/(RHCAV(I,J))
    WTH(I,J) = ROM(I,J)*WSFL-CFEM(I,J)*DTHCS(I,J)*WSURF(I,J)*DTHDT(I,J)
    IF (I.LT.ILE(J)) WTH(I,J) = LAMCAF(UOM(I,J),I,J)/ROM(I,J)-OMEGA*
    1-ROM(I,J)
    IF (I.LT.IE(J)) WTH(I,J) = VTH-(I,J)*ROM(I,J)-OMEGA*
    1-ROM(I,J)
    VTH(I,J) = WTH(I,J)-OMEGA*ROM(I,J)
    WSO = WTH(I,J)**2+WSURF(I,J)**2+WSURF(I,J)**2*WSURF(I,J)**2
    WTEMP = SORTWSO
    IF(WI(I,J).NE.O.) RELER = AMAX1(RELER,ARS((WTEMP-WI(I,J))/WI(I,J))
    WI(I,J) = WTEMP
    TWLMR = 2.*OMEGA*LAMDAF(UOM(I,J),I,J)-(OMEGA*ROM(I,J))**2/
    TIPF = 1./(WSO+TWLMR)**2/CP/TIPF(UOM(I,J))**2/2.*
    TF(TPTIP,LT.C.) GO TO 70
    RH0(I,J) = RHOPF(UOM(I,J))*TIPF**2*EXPON
    TPP = TPTPTIP(I)*TIPF(UOM(I,J))
    DPPR = DPPS(I,J)*SPHI(I,J)*CPR(I,J)*SPHI(I,J)
    DPIR = DPIS(I,J)*SPHI(I,J)*CPR(I,J)*SPHI(I,J)
   151
\[
\begin{align*}
XICMT &= (AR/PREL(J) \cdot DPDR - (OMIC + OMEGA \cdot ROM(I,J))) / PPP / CP \\
ZETOM &= OMEGA \cdot ROM(I,J) - AR/PREL(J) \cdot PPP \cdot DPDR \\
X1OM(I,J) &= XNEW \cdot XICMT + (1. - XNEW) \cdot XIOM(I,J) \\
Z1OM(I,J) &= ZNEW \cdot ZETOM + (1. - ZNEW) \cdot ZETOM(I,J)
\end{align*}
\]

60 CONTINUE
WRITE(NWRIT,1020) ITERRELER
C
C--ADJUST PRINTING CONTROL VARIABLES
C
IF (RELER.GE.VELTOL) RETURN
IF (RELER.EQ.0.) RETURN
IENC = IENC+1
IF (IMESH.GT.1) IMESH=1
IF (ISLINE.GT.1) ISLINE=1
IF (ISTATL.GT.1) ISTATL=1
IF (IPLOT.GT.1) IPLOT=1
IF (ITSON.GT.1) ITSON=1
IF (ICEBUG.GT.1) ICEBUG=1
RETURN
70 WRITE(NWRIT,1000) STOP
80 WRITE(NWRIT,1010) STOP
1000 FORMAT(68HPROGRAM STOPPED IN NEWRHQ DUE TO EXCESSIVE STREAM FUNCTION GRADIENT) 
1010 FORMAT(62H THE UPSTREAM INPUT WHIRL OR TANGENTIAL VELOCITY IS TOO LARGE)
1020 FORMAT(5X,9HITERATION,13,4H, MAXIMUM RELATIVE CHANGE IN VELOCITY = ,G11.4)
END

SUBROUTINE OUTPUT
C
C--OUTPUT CALCULATES AND PRINTS THE MAJOR OUTPUT DATA
C--AND ALONG STATION LINES FROM HUB TO SPROUD
C
COMMON SRW, SRE, ITER, IENC, NREAD, NWRITE, GAM, AR, MSFL, OMEGA, REDFAC, VELTOL, FNEW, DNEW, MBI, MBO, 1 MHT, NHT, NTIP, NTIN, NCUT, NLPL, NPPPC, NOSTAT, NSL, LSFR, 2 LTLT, LAMVT, IMESH, ISLINE, ISTATL, IPILOT, ISUPER, ITCIN, IDEBUG, 3 ZOMIN, ZOMBI, ZOMBO, ZOMOUT, ZHIN, ZTIN, ZHOUT, 7TCUT, 7HUB(50), 4 RHUN(50), ZTIP(50), RTIP(50), SFIN(50), RADIN(50), RALN(50), PRIP(50), 5 LAMIN(50), VTHIN(50), SFOUT(50), RADOUT(50), RACUT(50), LOSOUT(50), 6 LAMCUT(50), VTHOUT(50), ZHST(50), ZTST(50), FLFR(50), 7 ZBL(50, 50), ZTBL(50, 50), ZHBL(50, 50), ZTBL(50, 50), TNBL(50, 50), COMMON/MACON/MHMT, MHTP, CP, EXPON, TGRG, PITCH, CURVHI, CURVTI, 1 CURVHO, CURVTC, RHIN, RTIN, RHOUT, RTOUT, RLEH, RLET, RTEH, RTEHT, 2 ZLE(50), RLE(50), ZTE(50), RTE(50), ZLEOM(100), RLEOM(100), 3 SLEOM(100), TLEOM(100), ZTEOM(100), RTEOM(100), STEC(100), 4 TTHET(100), TLE(100), TLE(101), ZOM(100, 101), RCM(100, 101), 5 SOM(100, 101), TOM(100, 101), BTH(100, 101), OTHS(100, 101), 6 DTHDT(100, 101), PLOSS(100, 101), CPHEL(100, 101), SPHEL(100, 101), COMMON/MACON/MVARC/MVARC/100, 101, UOM(100, 101), K(100, 101), RHE(100, 101), 1 WSUBS(100, 101), WPSTIT(100, 101), WSUBT(100, 101), WSUBT(100, 101), 152
C--CALCULATE VELOCITY COMPONENTS AND FLOW ANGLES

DEGRAD = 180. / 3.1415927
DO 10 J = 1, MHTPI
  DO 10 I = 1, MM
    WSUBM(I,J) = SQRT(WSUBS(I,J)**2 + WSUBT(I,J)**2)
    SAMP(I,J) = WSUBT(I,J) / WSUBM(I,J)
    CAMP(I,J) = WSUBS(I,J) / WSUBM(I,J)
    WSUBZ(I,J) = WSUBS(I,J) * CPHI(I,J) - WSUBT(I,J) * SPI(I,J)
    WSS8(I,J) = WSUBZ(I,J) / WSUBM(I,J)
    ALPHA(I,J) = ASIN(WSUBS(I,J) * SPI(I,J))
    BETA(I,J) = ASIN(WSUBT(I,J) / WSUBM(I,J))
  10 CONTINUE

ENTRY TOUTPT
C--CALCULATE VELOCITY COMPONENTS AFTER TRANSONIC SOLUTION

DO 20 J = 1, MHTPI
  DO 20 I = 1, MM
    WSMR(M,I,J) = W(I,J) * COS(BETA(I,J))
    WMYM(I,J) = W(I,J) * SIN(BETA(I,J))
    WSM8(I,J) = WSMR(M,I,J) * COS(ALPHA(I,J))
    WSMR(I,J) = WSMR(M,I,J) * SIN(ALPHA(I,J))
  20 CONTINUE

C--CALCULATE STREAMLINE CURVATURE AND CRITICAL VELOCITY RATIO

DC 50 I = 1, MM
DO 40 J = 1, MHTPI
  TVERT(J) = TCM(I,J)
  ALVETT(J) = ALPHA(I,J)
  CALL SLOPES(TVERT, ALRET, MHTPI, DALVER)
  TVERT(J) = TVERT(TJ)
  DO 40 I = 1, MM
    DALDT(I,J) = DALVER(J)
    CALL SLOPES(SOM(1,J), ALPHA(1,J), MM, CALDS)
  40 CONTINUE

TPP(I,J) = TIPF(UOM(I,J)) - (2. * OMEGA * LAMDAF * UOM(I,J) / (1.J, I,J) - OMEGA * IROM(I,J)*2) / 2. / CP
IF (TPP.LE.0.) TPP=1.
60 WWCRI(I,J) = WLI(J)/SORT(TGROG*TPP)
C
C--COMPUTE PLATE SURFACE VELOCITIES BY STANITZ METHOD
C
CALL RLOVEL
C
C--CHECK IF UPPER OR LOWER SURFACE IS SUCTION SURFACE
C
MCT = 2
IF (((LAMDAF(.5, ILE(1), 1) - RVTHTA(.5, ILE(1), 1)) CMEGA.LT.0.) .AND. ((LAMDAF(.5, ILE(1), 1) - RVTHTA(.5, ILE(1), 1)).GT.0.) GO TO 80
DO 70 J=1,MHPTP
DC 70 I=1,MM
WDUM = WLSURF(I,J)
WLSURF(I,J) = WTSURF(I,J)
70 WTSURF(I,J) = WDUM
C
C--PRINT OUTPUT ROW BY ROW FROM HUB TO TIP
C
80 IF (IMESH.LE.0.) GO TO 100
IF (((ITER/IMESH)*IMESH NE.ITER .AND. ITER.NE.1) .OR. (ITER/IMESH).NE.ITER) GO TO 100
WRITE(NWRIT,1000)
IF (REDFAC.LT.1.0) WRITE(NWRIT,1150) ITER
IF (REDFAC.LE.1.0 AND IEND.LE.0) WRITE(NWRIT,1160) ITER
IF (REDFAC.LE.1.0 AND IEND.EQ.1) WRITE(NWRIT,1170)
IF (REDFAC.LE.1.0 AND IEND.EQ.2) WRITE(NWRIT,1180)
DO 90 J=1,MHTP1
WRITE(NWRIT,1010) J
WRITE(NWRIT,1020)
DO 90 I=1,MM
PHI = ARSIN(SPHI(I,J))DEGRAD
ALPHA(I,J) = ALPHA(I,J)DEGRAD
BETA(I,J) = BETA(I,J)/DEGRAD
WRITE(NWRIT,1030) I,J,ZOM(I,J),RCM(I,J),UOM(I,J),WSUBM(I,J),
1WTR(I,J),W(I,J),WWCR(I,J),ALPHA(I,J),BETA(I,J),PHI
ALPHA(I,J) = ALPHA(I,J)/DEGRAD
50 BETA(I,J) = BETA(I,J)/DEGRAD
C
C--INTERPOLATE TO OBTAIN OUTPUT DATA ON STREAMLINES
C
100 IF (ISLINE.LE.0.) GO TO 110
IF (((ITER/ISLINE)*ISLINE.EQ.ITER .OR. ITER.EQ.1) .OR. ((ITER/ISLINE).NE.ITER)) GO TO 110
110 IF (IPLLOT.LE.0.) GO TO 120
IF (((ITER/IPLLOT)*IPLLOT.EQ.ITER .OR. ITER.EQ.1) .OR. ((ITER/IPLLOT).NE.ITER)) GO TO 120
120 IF (ITSON.LE.0.) GO TO 220
IF (((ITER/ITSON) .AND. ITSON.NE.ITER)) GO TO 220
C
C--CALCULATE STREAMLINE ZSL,RSL COORDINATES FOR PRINT OUT
C
130 DC 150 I=1,MM
DO 140 J=1,MHTP1
ZTEM(J) = ZOM(I,J)
RTEM(J) = RCM(I,J)
140 UTEM(J) = UOM(I,J)
CALL SPLINT(UTEM,RTEM,MHTP1,FLFR,NSL,RSLTEM,AAA)
CALL SPLINT(RTEM,UTEM,MHTP1,RSLTEM,NSL,ZSLTEM,AAA)
DO 150 JS=1,NSL
ZSL(I,JS) = ZSLTEM(JS)
150 RSL(I,JS) = RSLTEM(JS)
C
C--CALCULATE STREAMLINE MLS COORDINATES FOR PRINTOUT AND PLOTTING

DO 170 JS=1,NSL
M(SL(1,JS)) = 0.
DO 160 IS=2,MM
160 M(SL(IS,JS)) = M(SL(IS-1,JS)) + 25 T(SL(IS,JS) - S(IS-1,JS))
1 + 25 T(SL(IS,JS) - S(IS-1,JS))
CALL SPLINT(ZSL(IS,JS),S(L(1,JS),MM,0.,1,ZEROM,SLREF)
DC 170 IS=1,MM
170 M(SL(IS,JS)) = M(SL(IS,JS)) - ZERO
C--INTERPOLATE TO OBTAIN OUTPUT DATA
II = 1
JJ = 1
DO 180 JS=1,NSL
DC 180 IS=1,MM
CALL LININT(ZOM,ROM,WSUBZ,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WZSL(IS,JS),II,JJ)
CALL LININT(ZOM,ROM,WSURR,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WRSL(IS,JS),II,II)
CALL LININT(ZOM,ROM,WT,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WTHSL(IS,JS),II,II)
CALL LININT(ZOM,ROM,WWCR,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WWCRSL(IS,JS),II,II)
CALL LININT(ZOM,ROM,CURV,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
ICURVSL(IS,JS),II,II)
W(SL(IS,JS)) = SORT(WZSL(IS,JS))**2 + WRSL(IS,JS)**2
ALPSL(IS,JS) = ATAN(WRSL(IS,JS)/WZSL(IS,JS))**EGRAD
BETSL(IS,JS) = ATAN(WTHSL(IS,JS)/WWCRSL(IS,JS))**DEGRAD
180 W(SL(IS,JS)) = SORT(WWCRSL(IS,JS))**2 + WTHSL(IS,JS)**2
C--CALCULATE ILS AND ITS ARRAYS OF STREAMLINE LOCATIONS INSIDE BLADE
C--LEADING AND TRAILING EDGES
C
CALL ILE TE
C--CALCULATE BLADE SURFACE VELOCITIES ON STREAMLINES BY INTERPOLATION
C
DC 190 JS=1,NSL
DO 190 IS=1,MM
WSSL(IS,JS) = 0.
190 WSSL(IS,JS) = 0.
II = 1
JJ = 1
DO 200 JS=1,NSL
ILLSJ = ILS(JS)
ITSJ = ITS(JS)
DC 200 IS=ILLSJ,ITSJ
CALL LININT(ZOM,ROM,WSURF,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WLSL(IS,JS),II,II)
200 CALL LININT(ZOM,ROM,WSURF,MM,MP1,100,101,SZ(IS,JS),RSL(IS,JS),
WTSVL(IS,JS),II,II)
C--PRINT OUTPUT ON STREAMLINES
C
IF (ISLINES.LE.0) GO TO 220
IF (ITER/ISLINES) ISLINES.LE.ITER.AND.ITER.NE.1) GO TO 220
WRITE(NWRT,1040)
WRITE(NWRT,1150) ITER
IF (READAC.LT.1.0) WRITE(NWRT,1160) ITER
IF (REDAC.EQ.1.0.AND.IEND.LE.0) WRITE(NWRT,1170) ITER
IF (READAC.EQ.1.0.AND.IEND.EQ.1) WRITE(NWRT,1180) ITER
DC 210 JS=1,NSL
155
WRITE(NWRT,1050) JS,FLFR(JS)
WRITE(NWRT,1060)
DC 210 IS=1,MM
WRITE(NWRT,1070) ZSL(IS,JS),RSL(IS,JS),MSL(IS,JS),WMSL(IS,JS),
1WTHSL(IS,JS),WSSL(IS,JS),WMCSRSL(IS,JS),ALPSL(IS,JS),BETSL(IS,JS),
2CURVSL(IS,JS),WLSSSL(IS,JS),WTSSL(IS,JS),ALPSL(IS,JS) = ALPSL(IS,JS)/DEGRAD
210 RETS(IS,JS) = BETSL(IS,JS)/DEGRAD

C
C--INTERPOLATE TO OBTAIN OUTPUT DATA ON HUB-SHROUD STATION LINES

C 220 IF (ISTATL.LE.0 .OR. NOSTAT.EQ.0) GO TO 410
   IF ((ITER/I STATL)*ISTATL.NE.ITER.AND.ITER.NE.1) GO TO 410
C--CALCULATE ZST AND RST ARRAYS
C--STORE HUB AND SHROUD POINTS INTO ZST AND RST ARRAYS
CALL SPLINT(ZHUB,RHUB,NHUB,2HST,NOSTAT,RTEM,AAA)
DC 230 IL=1,NOSTAT
ZST(1,IL) = 2HST(1)IL
230 RST(1,IL) = RTEM(1)IL
CALL SPLINT(ZTIP,RTIP,NTIP,ZTST,NOSTAT,RTEM,AAA)
DC 240 IL=1,NOSTAT
ZST(NSL,IL) = ZTST(1)IL
240 RST(NSL,IL) = RTEM(1)IL

C--CALCULATE INTERIOR POINTS IN ZST AND RST ARRAYS
DC 350 IL=1,NOSTAT
MARK(IL) = 1
RTEM(1) = RST(1,IL)
RTEM(20) = RST(NSL,IL)
DELR = (RTEM(20)-RTEM(1))19.0
ZTEM(1) = ZST(1,IL)
ZTEM(20) = ZST(NSL,IL)
DELZ = (ZTEM(20)-ZTEM(1))19.0
DO 250 J=2,19
250 RTEM(J) = RTEM(J-1)+DELR

C--CHECK FOR LEADING OR TRAILING EDGE STATION
   DELCH = (ZTEM(1) - ZLE(IL))1ZTEM(NBLPL) - ZLE(NBLPL))*C.05
   IF ((ZST(1,IL) .GT. (ZLE(IL)+DELCH)) .AND. (ZST(1,IL) .LT. (ZTE(IL)+DELCH))
      1.AND. (ZST(NSL,IL) .GT. (ZLE(NBLPL)+DELCH)) .AND. (ZST(NSL,IL) .LT. (ZTE(NBLPL)+DELCH))
      MARK(1L) = 2
   IF ((ZST(1,IL) .LT. (ZLE(IL)+DELCH)) .AND. (ZST(1,IL) .LT. (ZTE(IL)+DELCH))
      1.AND. (ZST(NSL,IL) .LT. (ZLE(NBLPL)+DELCH)) .AND. (ZST(NSL,IL) .LT. (ZTE(NBLPL)+DELCH))
      MARK(1L) = 3
   IF (MARK(1L) .LT. (ZLE(1)+DELCH)) .AND. (ZST(1,IL) .LT. (ZTE(1)-DELCH))
      MARK(1L) = 4
   IF (MARK(1L).EQ.2) GO TO 270
   IF (MARK(1L).EQ.3) GO TO 290
C--REGULAR STATION
DC 260 J=2,19
260 ZTEM(J) = ZTEM(J-1)+DELZ
GO TO 310
C--LEADING EDGE STATION
270 DO 280 JN=1,NBLPL
   ZLTEM(JN) = ZLE(JN)
280 RPLTEM(JN) = RLE(JN)
   CALL SPLINT(RPLTEM,ZLTEM,NBLPL,RTEM,20,ZTEM,AAA)
GO TO 310
C--TRAILING EDGE STATION
290 DO 300 JN=1,NBLPL
   ZLTEM(JN) = ZTE(JN)
300 RPLTEM(JN) = RTE(JN)
156
CALL SPLINT(RBLTEM,ZBLTEM,NBLPL,RTEM,20,ZTEM,AAA)

C--INTERPOLATE FOR STREAM FUNCTION
310  UTEM(I) = 0.
     UTEM(20) = 1.
     II = 1
     JJ = 1
     DO 320  J = 2, N2
320  CALL LININT(ZOM,ROM,WMUZ,MM,MHTTP1,100,101,ZTEM(J),RTEM(J),UTEM(J),
                     II, JJ)
C--CALCULATE STATION LINE RST COORDINATES FOR PRINT OUT
CALL SPLINT(UTEM,RTEM,20,FLFR,NSL,RST(1,IL),AAA)
     DELR = RST(NSL,IL)-RST(1,IL)
     DEL7 = ZST(NSL,IL)-ZST(1,IL)
     NSLM1 = NSL-1
C--CALCULATE STATION LINE ZST COORDINATES FOR PRINT OUT
IF (MARK(IL).EQ.2.OR.MARK(IL).EQ.3) GO TO 340
     DC 330  JL=2,NSLM1
330  ZST(JL,IL) = ZST(1,IL)+(RST(JL,IL)-RST(1,IL))/DELR*DELZ
     GO TO 350
340  CALL SPLINT(RBLTEM,ZBLTEM,NBLPL,RST(1,IL),NSL,ZST(1,IL),AAA)
     CONTINUE
C--CALCULATE STATION LINE MST COORDINATES FOR PRINT OUT
     DC 380  JL=1,NSL
     DO 360  IL=1,NSSTAT
             ITEMP(IL) = ZST(JL,IL)
360  RTEM(IL) = RST(JL,IL)
     MTFM(1) = 0.
     DC 370  IL=2, NSSTAT
370  WTEM(IL) = WTEM(IL-1)+SORT((ZTEM(IL)-ZTEM(IL-1))**2+(RTEM(IL)-
                     RTEM(IL-1))**2)
     CALL SPLINT(ZTEM,MTEM,NSSTAT,0.,1.,ZEROM,SLREF)
     DO 380  IL=1,NSSTAT
             MS1(JL,IL) = VTEM(IL)-ZEROM
380  CONTINUE
C--INTERPOLATE TO OBTAIN OUTPUT DATA ON STATION LINES
II = 1
JJ = 1
     DC 390  IL=1,NSSTAT
     DO 390  JL=1, NSL
             CALL LININT(ZOM,ROM,WSUBZ,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WSZST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,WSUBR,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WRSST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,WTH,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WTHST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,WWR,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WWRST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,WMV,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WMVST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,CURV,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WCURVST(JL,IL),II, JJ)
             WSST(JL,IL) = SQRT(WZST(JL,IL)**2+WRST(JL,IL)**2)
             ALPST(JL,IL) = ATAN(WRST(JL,IL)/WZST(JL,IL))*DGEGRAD
             BETST(JL,IL) = ATAN(WTST(JL,IL)/WSST(JL,IL))*CEGGRAD
             WST(JL,IL) = SORT(WSST(JL,IL)**2+WTST(JL,IL)**2)
             MLSST(JL,IL) = 0.
             WTSST(JL,IL) = 0.
             IF (MARK(IL).EQ.1) GO TO 390
             CALL LININT(ZOM,ROM,WSURF,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WLSST(JL,IL),II, JJ)
             CALL LININT(ZOM,ROM,WTST,MM,MHTTP1,100,101,ZST(JL,IL),RST(JL,IL),
                     WTSST(JL,IL),II, JJ)
390 CONTINUE
C--PRINT OUTPUT ALONG HUR-SHROUD STATION LINES
C
WRITE(NWRIT,1080)
IF (REDFAC.LT.1.0) WRITE(NWRIT,1150) ITER
IF (REDFAC.EQ.1.0.AND.IEND.LE.0) WRITE(NWRIT,1160) ITER
IF (REDFAC.EQ.1.0.AND.IEND.EQ.2) WRITE(NWRIT,1180)
IF (REOFAC.EQ.1.0.AND.IEND.LE.0) WRITE(NWRIT,1160) ITER
IF (REOFAC.EQ.1.0.AND.TEFD.EQ.1) WRITE(NWRIT,1170)
IF (REOFAC.EQ.1.0.AND.IEND.EQ.2) WRITE(NWRIT,1180)
WRITE(NWRIT,1130)
IF (MARK(IL).EQ.1) WRITE(NWRIT,1090) IL
IF (MARK(IL).EQ.2) WRITE(NWRIT,1100) IL
IF (MARK(IL).EQ.3) WRITE(NWRIT,1110) IL
IF (MARK(IL).EQ.4) WRITE(NWRIT,1120) IL
WRITE(NWRIT,1130)
IF (MARK(IL).EQ.1) WRITE(NWRIT,1090) IL
IF (MARK(IL).EQ.2) WRITE(NWRIT,1100) IL
IF (MARK(IL).EQ.3) WRITE(NWRIT,1110) IL
IF (MARK(IL).EQ.4) WRITE(NWRIT,1120) IL
WRITE(NWRIT,1130)
DO 400 JL=1,NSL
WRITE(NWRIT,1140) RST(JL,IL),ZST(JL,IL),MST(JL,IL),FLFR(JL),
1WST(JL,IL),WST(JL,IL),WWCRST(JL,IL),ALPST(JL,IL),
2RETST(JL,IL),CURVST(JL,IL),WLSSST(JL,IL),WTSST(JL,IL),
ALPST(JL,IL) = ALPST(JL,IL)/DEGCRAD
400 CONTINUE
C--CALCULATE DATA FOR INPUT TO THE TSONIC PROGRAM
C
100 IF (ITSON.LE.0) RETURN
IF (((ITER/ITSON)*ITSON.NE.ITER) RETURN
CALL TSONIN
RETURN
C--FORMAT STATEMENTS
C
1000 FORMAT (1H1////28r,79H*** STREAM FUNCTION, INTERIOR VELOCITIES, V
1ELocity COMPONENTS, AND ANGLES ***/44X,41HAT ALL MESH POINTS OF T
2HE ORTHOGONAL MESH/44X,41H(*)
1010 FORMAT (///42X,35H** HORIZONTAL ORTHOGONAL MESH LINE NO.,
112,3H//)
1020 FORMAT (1X,10HMEsch-POINT,3X,5HAXIAL,8X,6HRADIAL,6X,6HSTREAM,4X,
16HMERID.,3X,9HERL.TANG.,4X,4HERL.,3X,5HCRIT.VEh.,3X,6HERMI.D.,3X,
28HFLOW,3X,4HMEssH/1X,9HCOLM,7X,6HCOORD,7X,7X,
35HFUNCTION,5X,4HVEL.,6X,4HVEL.,7X,4HVEL.,5X,5HANGULAR,3X,5HANGLE)
1030 FORMAT (///,13H(*) (J),5X,3H(Z),10X,3H(R),10X,3H(U),6X,4H(WM),5X,5H(WTH),
7X,3H(W),5X,7H(W/WCR),3X,7H(ALPHA),3X,6H(BETA),5X,5H(PHI))
1040 FORMAT (1X,13X2X132X2X12G12.5,1X,F8.4,3(IX,F9.2),1X,F9.3,
13(3X,F7.2))
1050 FORMAT (1H1////15X,99H*** STREAM FUNCTION, INTERIOR VELOCITIES, V
1ELocity COMPONENTS, ANGLES, AND SURFACE VELOCITIES ***/56X,17HALO
2NG STREAMLINES/56X,17(1H*))
1060 FORMAT (///36X,20H** STREAMLINE NUMBER,13,23H -- STREAM FUNCTION
1=,F8.4,3H//)
1060 FORMAT (///,36X,20H** STREAMLINE NUMBER,13,23H -- STREAM FUNCTION
1=,F8.4,3H//)
1070 FORMAT (3F1X,G12.5,3(1X,F8.2),1X,F7.3,2(2X,F7.2),2X,G11.4,
1F8.2,2X,F8.2)
1080 FORMAT (1H1////15X,99H*** STREAM FUNCTION, INTERIOR VELOCITIES, V
IFLCITTY COMPONENTS, ANGLES, AND SURFACE VELOCITIES ***/28X,72HALO
2NG LINES FROM HUB TO SHROUD AT VARIOUS STATIONS THROUGH THE BLADE 3ROW/28X,77(1H*))
1090 FORMAT(/49X,26H** HUB-SHROUD STATION NO. ,12,3H ***/)
1160 FORMAT(/49X,26H** HUB-SHROUD STATION NO. ,12,3H ***/16X,
118H** LEADING EDGE ///</)
1130 FORMAT(/49X,26H** HUB-SHROUD STATION NO. ,12,3H ***/16X,
118H** WITHIN BLADE ///</)
1150 FORMAT(/53X,23(1H*)/53X,23H* REDUCED MASSFLOW */53X,23(1H*)/)
1160 FORMAT(/52X,25(1H*)/52X,25H* FULL MASSFLOW */52X,25(1H*)/)
1170 FORMAT(/52X,25(1H*)/52X,25H* FULL MASSFLOW */52X,25(1H*)/)
1180 FORMAT(/52X,25(1H*)/52X,25H* FULL MASSFLOW */52X,25(1H*)/)
END
SUBROUTINE BLOVEL
C--BLOVEL CALCULATES BLADE SURFACE VELOCITIES AND FR
C
COMMON SRW,SRE,ITER,IERC,NCFAIL,NWRIT
COMMON/INPUTT/GAMARMSFLOMARECFACVELTOL,FNEW,DNEW,MB1,MBO,
1 MM,MHT,NBL,NHBR,NTP,NIN,NOUT,NLBD,NP.HandlerFunc,LSFR/,2
1 LTPL,LAMVT,MESH,ISLINE,ISTATL,IPLOT,ISUPER,ITSON,IDEBUG,
1 ZMIN,ZMTB,ZMIN,ZTIN,ZTOUT,7TCUT,7HUB(50),
2 RMIN(50),ZTIP(50),RTIP(50),SFIN(50),RMIN(50),TIP(50),
2 LAMIN(50),VTHIN(50),SFCUT(50),RMIN(50),PRCP(50),
2 LAMOUT(50),VTHOUT(50),ZTST(50),ZTST(50),3FLFR(50),
2 7BL(50),ZBLY(50),RBL(50),THBL(50),TBL(50),
3 COMMON/CALC/SPHI(50,100),101),RHO(100,101),
4 COMMON/VARCOM/A(4,100,101),UOM(100,101),X(100,101),RHO(100,101),
C---CALCULATE DFCT
DO 30 I=1,MM
   DO 20 J=1,MHTPI
      TVERT(J) = TOM(I,J)
      FST(I,J) = VTH(I,J)*ROM(I,J)
      FVERT(J) = FST(I,J)
   CONTINUE
   CALL SLOPES(TVERT,FVERT,MHTPI,DFVERT)
   DO 30 J=1,MHTPI
      DFDT(I,J) = DFVERT(J)
   CONTINUE
C---CALCULATE DFDS, THEN DFDM AND BLADE SURFACE VELOCITIES
DO 40 J=1,MHTPI
   CALL SLOPES(SOM(I,J),FST(I,J),PM,DFDS)
   DO 40 I=1,MM
      DFDM(I,J) = -(DFDS(I)*CAMP(I,J)+DFDT(I,J)*SAMP(I,J))*BTH(I,J)*ICOS(RH(A(I,J)))
C---CALCULATE BLADE-TO-BLADE AVERAGE DENSITY
   TWLMR = 2.*OMEGA*LAMDAF(UOM(I,J),I,J)-(OMEGA*ROM(I,J))**2
   WSO = WLSURF(I,J)**2
   TTIP = 1.-((WSO+TWLMR)/CP/TIPF(UOM(I,J)))/2.
   IF(TTIP.LT.0.) TTIP = 0.
   RHOL = RHOIFP(UOM(I,J)) + TTIP**EXPON
   WSO = WTSURF(I,J)**2
   TTIP = 1.-((WSO+TWLMR)/CP/TIPF(UOM(I,J)))/2.
   IF(TTIP.LT.0.) TTIP = 0.
   RHOT = RHOIFP(UOM(I,J)) + TTIP**EXPON
   DELRHC(I,J) = RHOL-RHOT
   RHOAV(I,J) = (RHOL+4.*RHO(I,J)+RHOT)/6.
C---CALCULATE F-SUB-R FOR SUBROUTINE COEF
   FRT = W(I,J)/BTH(I,J)*(DTDS(I,J)*SPhI(I,J)+DFTCT(I,J)*CPHI(I,J))
   IFDM(I,J) = FCHANG = AMAX1(FCHANG,ABS(FRT-FRT(I,J)))
   FMAX = AMAX1(FMAX,FRT)
   FMIN = AMIN1(FMIN,FRT)
   FRT(I,J) = FNEW*FRT+(1.-FNEW)*FP(I,J)
40 CONTINUE
IF (IEND.LT.1) WRITE(NWRIT,1030) FMAX,FMIN,FCHANG
C---PRINT DEBUG OUTPUT IF REQUESTED
IF ([DEBUG] L.E.0) RETURN
IF (((ITER/.DEBUG)*IDEBUG .NE. 1) AND .ITER .NE. 1) RETURN
WRITE(NWRIT,1010)
WRITE(NWRIT,1000) ((I,J),WSUBS(I,J),WSUBT(I,J),VTH(I,J),RHO(I,J), RHOAV(I,J),DELRHC(I,J),DLDOU(I,J),I=1,MM),J=1,MHTPI)
SUBROUTINE ILETE

C-- ILETE CALCULATES THE INTEGER ARRAYS OF MESH POINT LOCATIONS WHICH ARE
C-- JUST INSIDE THE LEADING AND TRAILING EDGES OF THE BLADE

C
C COMMON/INPUTT/GAM,AR,MSPL,OMEGA,REDAC,VELTOL,FNEW,DNEW,MBI,MBO,
1 MM,HTN,NBL,NNUB,NTP,NIN,NCUT,NBLPL,NPPP,NOSTAT,NSL,LSFR,
2 LTPLE,AMVT,IMESH,ISLINE,ISTATL,IPL,ISUPER,ITSON,IDEBUG,
3 YOMIN,ZOMBI,ZOMBD,ZOMOUT,ZHIN,ZTIN,ZHOUT,ZTOUT,ZHUR(50),
4 RHUR(50),ZTIP(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTHIN(50),SFOUT(50),RADOUT(50),PRCP(50),LSSOUT(50),
6 LAMOUT(50),VTHOUT(50),TSPIN(50),TSPOUT(50),FLFR(50),
7 ZBL(50,50),RBL(50,50),THBL(50,50),TNBL(50,50)

COMMON/CALCON/MMMI,MHTP1,CP,EXPEN,TGROG,PITCH,CURVHI,CURVTI,
1 CURVHO,CURVTO,RTM,RTIN,RTOUT,REL,RLET,RTET,
2 TLE(50),RLE(50),TTE(50),RTE(50),ZLEOM(101),RLFOM(101),
3 SELMOM(101),TLEOM(101),ZTEOM(101),ZTEOM(101),STEO(101),
4 TLTEOM(101),LTEOM(101),ITEOM(101),ZCM(100,101),ROM(100,101),
5 SOM(100,101),TOM(100,101),RTH(100,101),DTHS(100,101),
6 DTHOT(100,101),PLOSS(100,101),CPHI(100,101),SPHI(100,101),

COMMON/SLOM/LSL(50),TSL(50),ZSL(100,50),RSL(100,50),MSL(100,50),
1 WZSL(100,50),WRSL(100,50),WMSL(100,50),WTHSL(100,50),
2 ALPSL(100,50),BLPSL(100,50),WLSL(100,50),WWCRSL(100,50),
3 CURVSL(100,50),WLSSL(100,50),WTSSL(100,50)

C-- LEADING EDGE
CALL SPLINT(RLE,ZLE,NBLPL,RLE(1),1,ZSPL,DZCR)
DO 20 J=1,NSL
   I = 0
10  I = I+1
   CALL SPLENT(RLE,ZLE,NBLPL,RSL(I),1,ZSPL,DZDR)
   IF (ZSPL.GT.ZSL(I,J)) GO TO 10
   20  ILS(I) = I
C-- TRAILING EDGE
CALL SPLINT(RTE,ZTE,NBLPL,RTE(1),1,ZSPL,DZDR)
DO 40 J=1,NSL
   I = ILS(J)-1
30  I = I+1
   CALL SPLENT(RTE,ZTE,NBLPL,RSL(I),1,ZSPL,DZDR)
   IF (ZSPL.GE.ZSL(I,J)) GO TO 30
SURREOTINE TSONIN

C--TSFt2NIN CALCULATES AND PRINTS OUT DATA AS INPUT TO THE
C--TSNC, r1AOE-TO-SLAlE ANALYSIS PROGRAM

COMMON SRW, SRE, ITER, IEND, NREA, NWRIT
COMMON/INPUT/GAM, AR, MSFL, OMEGA, RECFAQ, VELTOL, FNEW, DNEW, MBI, MBO,
1 MM, MHT, NBL, NHUB, NTIP, NIN, NOT, NBLPL, NPPP, NSTATE, NSL, LSFR,
2 LTPK, LAMVT, LINESH, ISLINE, ISTATL, IPILOT, ISUPER, ITSON, IDEBUG,
3 ZOMIN, ZOMBI, ZOMBO, ZOMOUT, ZHIN, ZOUT, ZTIN, ZOUTH, ZOUTH, ZHUB(50),
4 RHUB(50), RTIP(50), RTHIP(50), SFIN(50), RADIN(50), TIP(50), PRIP(50),
5 LAMIN(50), VTHIN(50), SFOUT(50), RADOUT(50), PRD(50), LOSOUT(50),
6 LAMOUT(50), VTHOUT(50), ZTHST(100), ZTST(100), FLFR(50),
7 ZBL(50, 50), RBL(50, 50), TBL(50, 50), TNNBL(50, 50)
COMMON/CALCN/MTF1, CP, EXPON, TPROC, PITCH, CURVHI, CURVTI,
1 CURWH0, CURVTO, RHIN, RTIN, RHOUT, RTOUT, RLEH, RLET, RTEH, RTEF,
2 ZLE(50), RLE(50), ZTE(50), RTE(50), ZLEOM(101), RLEOM(101),
3 TLEOM(101), TLE(101), TTE(101), ZOM(100, 101), RLE(100, 101),
4 TTHEOM(101), TTHE(101), TTHE(101), ZOM(100, 101), RLE(100, 101),
5 SCOM(100, 101), TM(100, 101), BMH(100, 101), TTHS(100, 101),
6 DTHS(100, 101), PLOS(100, 101), CPHI(100, 101), SPH(100, 101),
COMMON/VARCOM/A(4, 100, 101), UOM(100, 101), K(100, 101), RHOL(100, 101),
1 WSUB1(100, 101), WSB(100, 101), WSUB2(100, 101), WSUBR(100, 101),
2 WSUR1(100, 101), WTH(100, 101), WTH(100, 101), W1(100, 101),
3 TLEPH(100, 101), TLEPH(100, 101), TLEPH(100, 101), TLEPH(100, 101),
4 WLSRF(100, 101), WTSSRF(100, 101), WTSRF(100, 101), WTSRF(100, 101),
5 WPHN(100, 101), WDEL(100, 101), WFR(100, 101), WFR(100, 101),
6 XION(100, 101), ZETOM(100, 101), DLPLU(100, 101)
COMMON/SLCNO/LSL(100, 50), LSL(100, 50), RSL(100, 50), MSL(100, 50),
1 WSL(100, 50), WRSL(100, 50), WSL(100, 50), WSL(100, 50),
2 ALPSL(100, 50), EELSL(100, 50), EELSL(100, 50), WSSRL(100, 50),
3 CURVSR(100, 50), WTSSL(100, 50), WSSL(100, 50), WSSL(100, 50),
DIMENSION MSLP(100), THSP(100), THSP2(100), WRP(100), PSP(100),
1 BESP(100), DIST(50), DTSS(50), ANG(50), DTHRLF(50, 50),
2 MLESP(5), MLESP(5), MLESP(5), MLESP(5), MLESP(5),
3 REAL MSFL, MSP, MR, MSL, MLESP, MTEP,
4 INTEGER BLDAT, AANDK, ERSOR, STRFR, SLCRC, SURVL

C--PRELIMINARY CALCULATIONS
WRITE(NWRIT,1000)
WTFL = MSFL/100.
CRF = 0.001
DENRO = 0.001
NRSF = MM
RLDAT = 1
AANDK = 0
ERSOR = 0
STRFR = 2
SLCRC = 2
INTVL = 2
SURVL = 3
DC 20 JN=1, NBLPL
**DIST** = 0.
**DC 10 IN=2, NPPP**
10 **DIST(IN)** = **DIST(IN-1)**+**SORT**((**ZBL**(IN,JN)-**ZBL**(IN-1,JN))**2)+**RBL**(IN,JN)-**RBL**(IN-1,JN))**2)
**CALL SPLINE**(DIST,THBL(IN,JN),NPPP,CTCS,ANG)
**DO 20 IN=1, NPPP**
**ANG(IN)** = **ATAN**(**RBL**(IN,JN)**DTDS**(IN))
**20** **DRTBL**(IN,JN) = **TNBL**(IN,JN)/**COS**(ANG(IN))

---

**C--CALCULATE AND PRINT OUT TSONIC DATA ALONG EACH OF THE STREAMLINES**
**C DO 70 JS=1, NSL**
**II = 1**
**JJ = 1**
**TIPTEM = TIPF**(FLFR(JS))
**RHOIP = RHOP**(FLFR(JS))

---

**C--INTERSECTION OF STREAMLINE WITH BLADE LEADING AND TRAILING EDGES**
**CALL INRSCT**(ZSL(1,JS),RSL(1,JS),MM,ZLE,RLE,NBLPL,ZLESL,RLESL)
**CALL INRSCT**(ZSL(1,JS),RSL(1,JS),MM,ZTE,RTE,NBLPL,ZTESL,RTESL)

---

**C--INLET AND OUTLET FLOW ANGLES**
**CALL LININT**(ZOM,ROM,BETA,MM,MHTP1,100,101,ZLESL,RLESL,BETA,II,JJ)
**CALL LININT**(ZOM,ROM,BETA,MM,MHTP1,100,101,ZTESL,RTESL,BETO,II,JJ)
**BETA1 = BETA1*57.295780**
**BETA0 = BETA0*57.295780**

---

**C--CALCULATE STREAMSHEET LOCATION AND THICKNESS**
**DO 30 IS=1, MM**
**MR(IS) = MSL(IS,JS)-MSL(1,JS)**
**RMSF(IS) = RSL(IS,JS)**
**CALL LININT**(ZOM,ROM,RHO,MM,MHTP1,100,101,ZSL(IS,JS),RSL(IS,JS),
IrHCSL,II,JJ)
**CALL LININT**(ZOM,ROM,BTH,MM,MHTP1,100,101,ZSL(IS,JS),RSL(IS,JS),
IrTHSL,II,JJ)
**CALL LININT**(ZOM,ROM,PLOSS,MM,MHTP1,100,101,ZSL(IS,JS),RSL(IS,JS),
IrLOSS,II,JJ)
**30** **BESP(IS) = WTFL/(RHOSL*MMSL(IS,JS)*RSL(IS,JS)*BTHSL)*(1.-PLCSSL)

---

**C--CALCULATE BLADE SURFACE COORDINATES**
**II = 1**
**JJ = 1**
**NRPLTS = ITS(JS)-ILS(JS)+3**
**SPLNO1 = NRPLTS**
**SPLNC2 = NRPLTS**
**ILSJ = ILS(JS)**
**ITSJ = ITS(JS)**
**MSP(1) = 0.**
**DELM = SQRT((ZSL(ILSJ,JS)-ZLESL)**2+(RSL(ILSJ,JS)-RLESL)**2)**
**CALL LININT**(ZBL,RBL,THBL,NPPP,NBLPL,50,50,ZLESL,RLESL,1)
**IHTSL,II,JJ)**
**CALL LININT**(ZBL,RBL,DRTBL,NPPP,NBLPL,50,50,ZLESL,RLESL,1)
**IDRL,II,JJ)**
**DO 40 IS=ILSJ, ITSJ**
**MSP(IS) = MR(IS)-MR(ILSJ)+DELM**
**CALL LININT**(ZBL,RBL,THBL,NPPP,NBLPL,50,50,ZSL(IS,JS),RSL(IS,JS),
IHTSL,II,JJ)**
**CALL LININT**(ZPL,RBL,DRTBL,NPPP,NBLPL,50,50,ZSL(IS,JS),RSL(IS,JS),1)

---

*163*
10 RAI, IS, JS
DL = DRBL/RSL(IS, JS)/2.
THSPI(IS, JS) = THSL+THESL+DBL
THSP2(IS) = THSL+THESL-DBL
40 ISA = ISB+1
DELM = SQRT((ZTESL-ZSL(IS, JS))**2+(RTESL-RSL(IS, JS))**2)
MSP(NBLPTS) = MSP(NBLPTS-1)+DELM
CHORDF = MSP(NBLPTS)
CALL LININT(ZBL, RBL, THBL, NPPP, NBLPL, 50, 50, ZTESL, RTESL, IHTESL, II, JJ)
CALL LININT(ZBL, RBL, ORTHBL, NPPP, NBLPL, 50, 50, ZTESL, RTESL, IHTESL, II, JJ)
DRBL = DRBL/RTESL/2.
THSPI(NBLPTS) = THTESL+THESL+DBL
THSP2(NBLPTS) = THTESL-THLESL-COL
C--SHIFT STREAMSHEET MERIDIONAL COORDINATES TO ORIGIN AT BLADE
C--LEADING EDGE
DELM = MR(IS, JS)-MSP(2)
DC 50 IS=1, MM
50 MR(IS) = MR(IS)-DELM
C--CALCULATE SPECIAL ARRAYS OF LOCAL BLADE SURFACE R*THETA COORDINATES
C--AT LEADING AND TRAILING EDGES OF BLADE SECTION
NSPTS = 5
IF (NBLPTS.LT.5) NSPTS=NBLPTS
TLEREF = (THSPI(1)+THSP2(1))/2.
TTEREF = (THSPI(NBLPTS)+THSP2(NBLPTS))/2.
DC 60 I=1, NSPTS
J = NBLPTS-NSPTS+1
MLESP(I) = MSP(I)
CALL SPLINT(MR, RMSP, MM, MLESPI(I), 1, RLEP, DRDM)
R1LEPI(I) = RLEP*(THSPI(I)-TLEREF)
R1LEP2(I) = RLEP*(THSP2(I)-TLEREF)
M1ESP(I) = MSP(J)
CALL SPLINT(MR, RMSP, MM, MTESP(I), 1, RTEP, DRDM)
RTT1P(I) = RTEP*(THSPI(J)-TTEREF)
60 RTT2P(I) = RTEP*(THSP2(J)-TTEREF)
C--PRINT TSONIC DATA
WRITE(NWRIT,1010) JS, FLFRA(JS)
WRITE(NWRIT,1020) JS
WRITE(NWRIT,1160) GAM, AR, TITEM, RHO, WTFL, OMEGA, ORF
WRITE(NWRIT,1030) JS
WRITE(NWRIT,1170) BETA, BETA0, CHORDF
WRITE(NWRIT,1040) JS
WRITE(NWRIT,1170) REDFAC, DENTCL
WRITE(NWRIT,1050) JS
WRITE(NWRIT,1180) NAR, NSP
WRITE(NWRIT,1060) JS
WRITE(NWRIT,1190) SPLNO1
WRITE(NWRIT,1070) JS
WRITE(NWRIT,1170) (MSP(I), I=1, NBLPTS)
WRITE(NWRIT,1080) JS
WRITE(NWRIT,1170) (THSPI(I), I=1, NBLPTS)
WRITE(NWRIT,1090) JS
WRITE(NWRIT,1190) SPLNO2
WRITE(NWRIT,1100) JS
WRITE(NWRIT,1170) (MSP(I), I=1, NBLPTS)
WRITE(NWRIT,1110) JS
WRITE(NWRIT,1170) (THSP2(I),I=1,NALPTS)
WRITE(NWRIT,1120)
WRITE(NWRIT,1170) (MR(I),I=1,MM)
WRITE(NWRIT,1130)
WRITE(NWRIT,1170) (RMSP(I),I=1,MM)
WRITE(NWRIT,1140)
WRITE(NWRIT,1120) (BESP(I),I=1,MM)
WRITE(NWRIT,1150)
WRITE(NWRIT,1200) BLOAT, AANDK, ERSOR, STRFN, SLCRC, INTVL, SURVL
C
C--PRINT SPECIAL LOCAL ARRAYS OF BLADE SURFACE DATA
WRITE(NWRIT,1210)
WRITE(NWRIT,1170) (MLESP(I),I=1,NSPTS)
WRITE(NWRIT,1120)
WRITE(NWRIT,1220)
WRITE(NWRIT,1170) (RTLEP1(I),I=1,NSPTS)
WRITE(NWRIT,1120)
WRITE(NWRIT,1170) (RTLEP2(I),I=1,NSPTS)
WRITE(NWRIT,1120)
WRITE(NWRIT,1170) (RTEP1(I),I=1,NSPTS)
WRITE(NWRIT,1120)
WRITE(NWRIT,1170) (RTEP2(I),I=1,NSPTS)
70 CONTINUE
RETURN
C
C--FORMAT STATEMENTS
C
1000 FORMAT (1H1///41X,39(1H*)/41X,39H*** INPUT DATA FOR TSONIC PROGRAM
IM ***/41X,39(1H*///)
1010 FORMAT (4X,17HSTREAMLINE NUMBER,13,23H -- STREAM FUNCTION =, 1F8.4/)
1020 FORMAT (7X,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRHOIP,12X,4HWTFL,27X, 15HOMEGA,12X,3HDORF)
1030 FORMAT (6X,5HBETAI,10X,5HBETAC,11X,6MCHORDF,11X,5HSTCRF)
1040 FORMAT (6X,6HRECFAC,10X,6HOENTOL)
1050 FORMAT (6X,8HMBI,18X,18HMMB NBB1 NBB2 NRS)
1060 FORMAT (7X,3HR1,12X,3HR2,12X,9VBET1,11X,5HBE21,11X,6HSPLNC1)
1070 FORMAT (7X,4HMSPI,2X,5HARRAY)
1080 FORMAT (7X,5HTSP1,2X,5HARRAY)
1090 FORMAT (7X,5HSTPI1,12X,5HSTPI2,12X,5HSTCP1,11X,6HSPLNC2)
1100 FORMAT (7X,5HSTSP1,2X,5HARRAY)
1110 FORMAT (7X,5HSTSP2,2X,5HARRAY)
1120 FORMAT (7X,9HMR ARRAY)
1130 FORMAT (7X,11HRMSP ARRAY)
1140 FORMAT (7X,11HRESP ARRAY)
1150 FORMAT (5X,47HBLOAT AANDK ERSOR STRFN SLCRC INTVL SURVL)
1160 FORMAT (1X,5G16.7,16X,2G16.7)
1170 FORMAT (1X,8G16.7)
1180 FORMAT (30X,215)
1190 FORMAT (65X,6G16.7)
1200 FORMAT (1X,717)
1210 FORMAT (1X,5HMLESP,2X,5HARRAY)
1220 FORMAT (7X,6HRTLEP1,2X,5HARRAY)
1230 FORMAT (7X,6HRTLEP2,2X,5HARRAY)
1240 FORMAT (7X,5HTESP,2X,5HARRAY)
1250 FORMAT (7X,6HRTTEP1,2X,5HARRAY)
1260 FORMAT (7X,6HRTTEP2,2X,5HARRAY)
SUPROUTINE INDEV
C
C--INDEV CALCULATES A CORRECTION TO CTMDS TO ALLOW FCP INCIDENCE AND
C--DEVIATION (AFTER BLOCKAGE CORRECTION)
C
COMMOM SRW,SRE,ITER,ITAP,NREAC,NWRIT
COMMON/INPUT/GAM,AR,MSL,OMEGA,REDFA,VELTOL,FNEW,DNEW,MRB,MBO,
1 MM,WHF,NBL,NHUB,NTIP,NIN,NOUT,NBLPL,NPPP,USTAT,NSL,LSR,
2 LTPL,IMESH,ISLINE,ISTATL,IFLCT,ISUPER,ISCON,IDEBUG,
3 ZOM1,ZOM2,ZOM3,ZOM4,ZHIN,ZTHIN,ZHOUT,ZTOUT,DTOUT,7HUB(50),
4 RHUB(50),RTIP(50),STIP(50),SFOUT(50),RADOUT(50),PRCP(50),LOSOUT(50),
5 LAMIN(50),VTHIN(50),SFOUT(50),RADOUT(50),PRCP(50),LOSOUT(50),
6 LAMOUT(50),VTHOUT(50),ZHST(50),ZTST(50),FLER(50),
7 ZRL(50,50),RLR(50,50),THRL(50,50),TNBL(50,50),
COMMOM/CALCON/MMW1,MMTP1,CP,EXPON,TGROG,PITCH,CURVHI,CURVT,
1 CURVHO,CURVTO,RHIN,RTHIN,RHOUT,RTOU,RLEH,RLTE,RTEH,RTET,
2 ZLE(50),RLE(50),ZTE(50),RTE(50),ZLEOM(101),RLEOM(101),
3 SLRAM(101),THLEOM(101),ZTEOM(101),RTEOM(101),STEOM(101),
4 THTEOM(101),TLE(101),ITE(101),ZOM(100,101),ROM(100,101),
5 SOM(100,101),TCH(100,101),ETH(100,101),DTMDS(100,101),
6 DTWQ(100,101),PLOSS(100,101),PHI(100,101),SBI(100,101),
COMMOM/VARCOW/A4,100,101),UGM(100,101),K(100,101),RHD(100,101),
1 WLSUB1(100,101),WLSUB2(100,101),WLSUB3(100,101),WLSUB(100,101),
2 WLSUB(100,101),WTH(100,101),VTH(100,101),WT(100,101),
3 ALPH(100,101),BETA(100,101),WWW(100,101),CURV(100,101),
4 WLSURF(100,101),WTSURF(100,101),CTMDS(100,101),CBLM(100,101),
5 RHOAV(100,101),DHLRS(100,101),FR(100,101),DFDQ(100,101),
6 XIOMO(100,101),ZETOM(100,101),DDLU(100,101),
COMMOM/INDCOM/7PC(11,50),PRPC(11,50),DTWQD(11,50),DTHDR(11,50)
DIMENSION OTDSE(101),DTDSTE(101)
DEGRAD = 180./3.1415927
IT = 1
JJ = 1
IID = 1
JJP = 1
IF (IMESH.LE.0) GO TO 10
IF ((ITER/IMESH)*IMESH.EQ.ITER.OR.ITER.EQ.1) GO TO 30
10 IF (ISLINE.LE.0) GO TO 20
IF ((ITER/ISLINE)*ISLINE.EQ.ITER.OR.ITER.EQ.1) GO TO 30
20 IF (ISTATL.LE.0) GO TO 40
IF ((ITER/ISTATL)*ISTATL.NE.ITER.AND.ITER.NE.1) GO TO 40
30 WRITE(NWRIT,1010)
IF (REDFA.LT.1.0) WRITE(NWRIT,1100) ITER
IF (REDFA.EQ.1.0 AND IFND.LE.0) WRITE(NWRIT,1110) ITER
IF (REDFA.EQ.1.0 AND ITEND.EQ.1) WRITE(NWRIT,1120) ITER
IF (REDFA.EQ.1.0 AND ITEND.EQ.2) WRITE(NWRIT,1130) ITER
WRITE(NWRIT,1020)
40 DO 120 J=1,MMTP1
C
C--CCRECT CTMDS FOR INCIDENCE AT BLADE LEADING EDGE
C
I = ILE(J)-1
EXTRAP = SLEOM(J)-SOM(I,J)
RTFSE = BETA(I,J)+EXTRAP*(BETA(I,J)-BETA(I-1,J))/(SOM(I,J)-ISOM(I-1,J))
CALL LININT(ZCM,ROM,BTH,MM,MHTP1,100,101,ZLEOM(J),RLEOM(J),BTHLE,111,J)
TANBL = TAN(RTFSE)*BTHLE/PITCH
SPHILE = SPHI(I,J)+EXTRAP*(SPHI(I+1,J)-SPHI(I,J))/(SOM(I+1,J)-ISOM(I,J))
CPhILE = CPHI(I,J)+EXTRAP*(CPHI(I+1,J)-CPHI(I,J))/(SOM(I+1,J)-ISOM(I,J))
ALPHLE = ALPHA(I,J)+EXTRAP*(ALPHA(I+1,J)-ALPHA(I,J))/(SOM(I+1,J)-ISOM(I,J))
CALL LININT(ZPC,RPC,DTHDZ,11,NDLPL,11,50,ZLEOM(J),RLEOM(J),DTCZLE,111,J)
CALL LININT(ZPC,RPC,DTHDZ,11,NDLPL,11,50,ZLEOM(J),RLEOM(J),DTCZLE,111,J)
DTDLTE = DTDLRE*CPHILE-DTCZLE*SPHILE
IF(ITER.EQ.0) DTDSLE(J) = DTDLRE*SPHILE+DTCZLE*CPHILE
TANBL = RLEOM(J)*(DTDLRE*SIN(ALPHLE)+DTCZLE*COS(ALPHLE))
BTBLD = ATAN(TANBL)
RLINC = (ATAN(TANRFL)-BTBLD)*DEGRAC
UBLNC = (BTFSLE-BTBLD)*DEGRAC
EXTRAP = SOM(I+1,J)-SLEOM(J)
SAMPLE = SAMP(I+1,J)+EXTRAP*(SAMPI+1,J)-SAMPI,J)/(SOM(I+2,J)-ISOM(I+1,J))
CAMPLE = CAMP(I+1,J)+EXTRAP*(CAMP(I+1,J)-CAMP(I+2,J))/(SOM(I+2,J)-ISOM(I+1,J))
DTDSFL = (TANBFL/RLEOM(J)-DTDLRE*SAMPLE)/CAMPLE
BLOCRD = (RLEOM(J)+RTOM(J))/2*(BTHLEOM(J)-HTH(OM(J))
BLOCRD = SORT(BLOCRD**2+(STEM(J)-SLECM(J))^2)
SLIDE = BLOCRD/PITCH/RLEOM(J)
DISTLE = AMIN(0.5,AMAX(1.6,(11-4*SLIDE)/18.))*(STEOM(J)-SLEOM(J))
I = ILE(J)
50
SCIST = SLECM(J)+DISTLE-SOM(I,J)
IF(SCDIST.LE.0) GO TO 60
DTDS(J,J) = DTDS(J,J)+(DTDSFL-DTDSLE(J))*SDIST/DISTLE
I = I+1
GO TO 50
60

50 SCIST = SLECM(J)+DISTLE-SOM(I,J)
IF(SCDIST.LE.0) GO TO 60
DTDS(J,J) = DTDSFL
C
C--CORRECT DTDS FOR DEVIATION AT BLADE TRAILING EDGE
C
I = ILE(J)+1
EXTRAP = SOM(I,J)-STEOM(J)
RTFSTE = BETA(I,J)+EXTRAP*(BETA(I,J)-BETA(I+1,J))/(SOM(I+1,J)-ISOM(I,J))
CALL LININT(ZCM,ROM,BTH,MM,MHTP1,100,101,ZTEOM(J),RTEOM(J),BTHTE,111,J)
CALL LININT(ZCM,ROM,PLOSS,MM,MHTP1,100,101,ZTEOM(J),RTEOM(J),PLOST,E,111,J)
TANBL = TAN(RTFSTE)*BTHTE/PITCH/(1-PLOSTE)
SPHITE = SPHI(I,J)+EXTRAP*(SPHI(I-1,J)-SPHI(I,J))/(SOM(I,J)-ISOM(I-1,J))
CPhITE = CPHI(I,J)+EXTRAP*(CPHI(I-1,J)-CPHI(I,J))/(SOM(I,J)-ISOM(I-1,J))
ALPHITE = ALPHA(I,J)+EXTRAP*(ALPHA(I-1,J)-ALPHA(I,J))/(SOM(I,J)-ISOM(I-1,J))
CALL LININT(ZPC,RPC,DTHCZ,11,NDLPL,11,50,ZTEOM(J),RTEOM(J),DTCZTE,
CALL LINT(ZPC,RPC,DTD+OR,11,NBLPL,11,50,ZTEOM(J),RTECM(J),DTDRT,E,11,J,J)

DTDRT = DTDRT+*C(PHILO+DZT+*S(PHILO
IF(ITER.LE.1) DTDST(E,J) = DTDRT+*S(PHILO+DZT+*C(PHILO)
TANRL = RTEOM(J)*(DTDRT+SIN(ALPHOM)+DZT+*COS(ALPHOM))
PTABL0 = ATAN(TANRL)
BLDEV = (ATAN(TANRL)-PTABL0)*CEGRAD
URDE = (BTDST(E-PTABL0)*DEGRAD
IF (IMESH.LE.0) GO TO 70
70 IF (ISLINE.EQ.ITER.OR.ITER.EQ.1) GO TO 90
90 WRITE(NWRIT,1000) J,BLINC,URINC,PLDEV,URDEV
100 EXTRAP = SLEOM(I,J)-SOM(I-1,J)
SAMPTE = SAMP(I-1,J)+EXTRAP*(SAMP(I-1,J)-SAMP(I-2,J))/(SOM(I-1,J)-
ISCM(I-2,J))
CAMPTE = CAMP(I-1,J)+EXTRAP*(CAMP(I-1,J)-CAMP(I-2,J))/(SOM(I-1,J)-
ISCM(I-2,J))
DTDSFL = (TANRL/RTEOM(J)-DTDOTTE*SAMPTE)/CAMPTE
SLIDTE = RLDGRD/PITCH/RTEOM(J)
DISTTE = AMN1(5,AMAX1(1/6.,11.-4.)*SLIDTE)/180)*RTEOM(J)-
ISLEOM(I,J)
I = ITER(J)
110 EDIST = SOM(I,J)-STEOM(J)+DISTTE
IF(SDIST.LE.0.) GO TO 120
DTHDSI,J) = DTHDSI,J)+(DTDSFL-DSTTE(J))*SOM/EDIST
I = I-1
GO TO 110
120 DTDST(E,J) = DTDSFL
WRITE(NWRIT,1140)
RETURN

C--FCRMAT STATEMENTS
C
1000 FORMAT (35X,1H*,2X,13,3X,1H*,F9.2,2X,F9.2,4X,1H*)
1010 FORMAT (1H*,44X,40*,*** INCIDENCE AND DEVIATION ANGLES ***/
15X,301(H*))
1020 FORMAT (/35X,10*, MESH *,8X,9HINCIDENCE,7X,1H*,8X,9HDEVIATION, 
17X,1H*,35X,10*, LINE *,3X,7HBLOCKED,3X,9HUNBLOCKED,2X,1H*,3X, 
27HBLOCKED,3X,9HUNBLOCKED,2X,1H*)
1100 FORMAT (/53X,231H*,53X,23H*, REDUCED MASSFLOW */53X,23(1H*)/
153X,18H*, ITERATION NO.,12,3H, */53X,23(1H*)])
1110 FORMAT (/52X,251H*,52X,25H*, FULL MASSFLOW */52X,251H*)/
152X,19H*, ITERATION NO.,12,4H, */52X,251H*)]
1120 FORMAT (/52X,251H*,52X,25H*, FULL MASSFLOW */42X,451H*)/
142X,1H*,12X,19HTRANSONIC SOLUTION,12X,1H*,/42X,45H* BY VELOCITY G
2RADIENT APPROXIMATE METHOD */35X,591H*)/35X,59H* ALL VELOCITIES 
3 SMALLER THAN CHOIKING MASSFLOW SOLUTION */35X,591H*)]
1130 FORMAT (/52X,251H*,52X,25H*, FULL MASSFLOW */42X,451H*)/
142X,1H*,12X,19HTRANSONIC SOLUTION,12X,1H*,/42X,45H* BY VELOCITY G 
2RADIENT APPROXIMATE METHOD */35X,591H*)/35X,59H* ALL VELOCITIES 
3 SMALLER THAN CHOIKING MASSFLOW SOLUTION */35X,591H*)]
1140 FORMAT (1H1)
END
SUBROUTINE SLPlot
C
C---SLPlot plots the streamlines in the hub-shroud flow plane
C
COMMON SRF, SRE, ITER, IEND
COMMON /INPUT/GAM, AR, MSFL, OMEGA, REDFAO, VELTCL, FNEW, CNEW, MBA, MOB,
  1 MM, MHT, RNL, NRU, NTIP, PIN, NCUF, NLPL, NPPP, NOSTAT, NSL, LSFR,
  2 LTP, LAMVT, IREMSL, ISLINE, ISTATL, IPLOT, ISIPER, ITSCN, ICEBUG,
  3 ZOMIN, ZOMBI, ZOMBO, ZOMOUT, ZHIN, ZTIN, ZHOUT, ZTCUT, ZHUB50,
  4 RHRCL50, ZTPI50, RTIP50, SFIN50, RADIN50, TIP50, PRIP50,
  5 LAMIN50, VTHIN50, SFOUT50, RADOUT50, PRCP50, LOSCUT50,
  6 LAMOUT50, VTHOUT50, HST50, ZST50, FLT50, FLFR50,
  7 ZAL5050, RBL5050, THPL5050, TNBL5050
COMMON /SLCOM/ TITL550), ITSL550), SL510050), RSL510050), MSL510050),
  1 WSSL510050), WRSCL510050), WMSL510050), WTHSL510050),
  2 ALPSL510050), BESL510050), WSL510050), WWCRSL510050),
  3 CURVSL510050), WLSSL510050), WTSSL510050)
COMMON /PLTCCM/ ZLNG, ZZRNG, RBRRNG, RTRNG, ZHPLT100), RHPLT100),
  1 ZPLT100), RSPLT100), ZLPLT100), RLP100), ZTPT5100),
  2 RPTL5100)
DIMENSION TITL110), TITL23), TITL33), TITL411), TITL55)
REAL MSL
DATA TITL1/'STRE', 'AMLIO', 'NE P', 'LOT$, 'C1SL', '2IN ', 'MERT', 'DION'
  1,'AL P', 'LANE'/
DATA TITL2/'7 D', 'IREC', 'TION'/
DATA TITL3/'R D', 'IREC', 'TION'/
DATA TITL4/'SUBS', 'OINC', 'SCL', 'OLUT', 'IONS', 'C2IT', 'ERAT', 'IONS'
  1,'CIN0', 'X', 'XXX'/
DATA TITL5/'TRAN', 'SONI', 'C$C1', 'SOLU', 'TION'/
DATA SYM/'X'/
IF (IPLOT.LE.0) RETURN
IF (ITER.IEQ.IPLOT) RETURN
IF (IEND .LE. 0) RETURN
IF (ITER .GE. IEND) RETURN
CALL LRGRIC11,0,0,0,0)
CALL LRRCNVT(ITER,1,TITL411,140)
IF (IEND.LE.0) CALL LRLEGNT(144,0,4,2,6,0,1,0)
IF (IEND.GT.0) CALL LRLEGNT(5520,0,4,2,5,5,1,0)
C
C---PLOT THE ITERATION NUMBER
CALL LRMRGN11,0,1,2,0,1,0)
CALL LRANGE(ZLNG, ZZRNG, RBRRNG, RTRNG)
CALL LRGRIC(-11,1,1,1,1,0)
CALL LRLEGNT(TITL540,0,3,5,0,7,0,0)
CALL LRCHS7(2)
CALL LRLEGNT(TITL212,0,4,5,1,5,0,0)
CALL LRLEGNT(TITL312,0,4,5,0,0)
CALL LRCHS7(4)
CALL LRCURV(ZHPLT, RHPLT, 100, 2, SYM, 0,0)
CALL LRCURV(ZSPLT, RSPLT, 100, 2, SYM, 0,0)
CALL LRCURV(ZPLT, RLP100, 2, SYM, 0,0)
CALL LRCURV(ZTPT100, ZTPT5100, SYM, 0,0)
C
C---PLOT STREAMLINES
ECP = 0.0
NSLI = NSL-1
DO 10 JS=2,NSLI
IF (JS.EQ.NSLL) EDP=1.0
10 CONTINUE
169
10 CALL LRCURV(7SL(1,JS),RSL(1,JS),MM,2,SYM,ECP)
   CALL LRCURV(7SL,RSL,0,1,SYM,1,C)
   RETURN
END

SUBROUTINE SVPLOT

C
C--SVPLOT PLOTS THE MEAN STREAM SURFACE AND BLADE SURFACE OUTPUT
C--VELOCITIES ALONG ALL STREAMLINES

C
COMMON SRW,SRFL,ITER,EEND
COMMON/INPUT/GAM,AR,MSFL,OMEGA,REDAC,VELTOL,FNEW,DONE,MBI,MBO,
1 MM,MHT,NML,NHUP,NTIP,NIN,NOUT,NRLPL,NPPP,NSTAT,NSL,LSFR,
2 LITPL,LAMVT,IMESH,LSLINE,ISATL,IPLCT,ISUPER,ITSCN,IDEBUG,
3 ZMIN,ZMBOI,ZMBOZ,ZHIN,ZTIN,ZHOUT,ZTOUT,ZHUB(50),
4 RHUR(50),ZTIP(50),ATIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTMIN(50),SFOUT(50),RADOUT(50),PRPF(50),LOSOUT(50),
6 LAMOUT(50),VTOUT(50),ZHST(50),7TST(50),F LFR(50),
7 ZRL(50,50),ZRRL(50,50),TBL(50,50),TNBL(50,50),
COMMON/SL COM/LSL(50),TSL(50),7SL(100,50),RSL(100,50),MSL(100,50),
1 W7SL(100,50),WRSL(100,50),HMSL(100,50),WTHSL(100,50),
2 ALPSL(100,50),RESL(100,50),WSL(100,50),WCRS(100,50),
3 CURVSL(100,50),MCLS(100,50),WTS(100,50),
DIMENSION TITL1(12),TITL2(9),TITL3(14),TITL4(15),
1 TITL5(16),TITL6(6),TITL7(2)
REAL MSL,LRNG
DATA TITL1/'MER1','DION','AL A','ND S','URFA','CE$C','1411','RELA',
1,'TIVE','VEL','OCIT','IES'/
DATA TITL2/'ST','REAM','LINE','NO.','','XXXX','''U = ''','XXXX'
1,'XXXX'/
DATA TITL3/'MER1','DION','AL R','RELAT','IVE ','VELC','CITI','ES$C'
1,1S$6','FOR ','ALL ','STRE','AML','INES'/
DATA TITL4/'SUCT','ION ','SURF','ACE ','RELAT','IVE ','VEL','OCIT'
1,'TES$','C1$','8FOR','AL ','STR','EAM','INES'/
DATA TITL5/'PRES','SURE ','SURF','ACE ','RELAT','IVE ','VEL ','OCIT'
1,'TIES$','C1$','8FOR','AL ','STR','EAM','INES'/
DATA TITL6//'ME ','RIDI','ONAL ','CO ','ORDI','NAYE'/
DATA TITL7//'VELO ','CITY'/
DATA SYM''X'/
IF (IPLT.LE.0) RETURN
IF ((ITER/IPLT).NE.ITER.AND.ITER.NE.1) RETURN

C--COMMON RANGE OF PLOTS AND SET UP FOR PLOTTING
C
LRNG = MSL(1,1)
RRNG = MSL(1,1)
BRNG = 1000.
TRNG = 0.
DC 30 JS=1,NSL
LRNG = AMAX(LRNG,MSL(1,JS))
RRNG = AMAX(RRNG,MSL(MM,JS))
ILSJ = ILS(JS)
ITSJ = ITSL(JS)
DC 10 IS=ILSJ,ITSJ
BRNG = AMAX(BRNG,WMSL(IS,JS))
BRNG = AMAX(BRNG,WTS(IS,JS))
TRAN = AMAXI(TRNG, WLSSL(IS, JS))
10 TRAN = AMAXI(TRNG, WTSSL(IS, JS))
DO 20 IS=1, MM
RRNG = AMIN1(TRAN, WSL(IS, JS))
20 TRAN = AMAXI(TRNG, WSL(IS, JS))
CONTINUE
CALL LRMRGN(1.0, 1.0, 2.0, 1.0)
CALL LRANGE(LRNG, RRNG, BRNG, TRAC)
CALL LRGRID(1, 1, 11.0, 11.0)
C
C--FLOT VELOCITIES ON EACH STREAMLINE
C
DC 40 JS=1, NSL
ILSJ = ILS(IS)
MPLD = ITS(IS) - ILS(IS) + 1
IF (JS.EQ.1) CALL LRLEGN(TITL1, 48, 0, 2.5, 0.7, 0.0)
CALL LRCHSZ(3)
CALL LRCHTV(JS, 1, TITL2, 51, 1, 4, 0)
CALL LRCHTV(TLFR(IS), 3, TITL2, 8, 3, 8, 4)
CALL LRLEGN(TITL2, 36, 0, 2, 2.9, 5.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL6, 24, 0, 3, 4, 1, 3, 0.0)
CALL LRLEGN(TITL7, 8, 1, 0, 2, 4, 9.0)
CALL LRCHS7(4)
CALL LRGCURV(MSL(1, JS), WSL(1, JS), MM, 2, SYM, 0.0)
CALL LRGCURV(MSL(1, JS), WSL(1, JS), MM, 4, SYM, 0.0)
CALL LRGCURV(MSL(ILS, JS), WLSSL(ILS, JS), MBLD, 2, SYM, 0.0)
CALL LRGCURV(MSL(ILS, JS), WLSSL(ILS, JS), MBLD, 4, SYM, 0.0)
CALL LRGCURV(MSL(ILS, JS), WTSSL(ILS, JS), MBLD, 2, SYM, 0.0)
CALL LRGCURV(MSL(ILS, JS), WTSSL(ILS, JS), MBLD, 4, SYM, 1.0)
C
C--PLOT MERIDIONAL VELOCITIES FOR ALL STREAMLINES
C
CALL LRGRID(3, 3, 11.0, 11.0)
CALL LRLEGN(TITL3, 56, 0, 1.7, 0.7, 0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL6, 24, 0, 3, 4, 1, 3, 0.0)
CALL LRLEGN(TITL7, 8, 1, 0, 2, 4, 9, 0.0)
CALL LRCHS7(4)
EOP = 0.0
DO 50 JS=1, NSL
IF (JS.EQ.1) EOP=1.0
50 CALL LRGCURV(MSL(1, JS), WSL(1, JS), MM, 2, SYM, EOP)
C
C--PLOT SUCTION SURFACE VELOCITIES FOR ALL STREAMLINES
C
CALL LRLEGN(TITL4, 60, 0, 1.2, 0.7, 0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL6, 24, 0, 3, 4, 1, 3, 0.0)
CALL LRLEGN(TITL7, 8, 1, 0, 2, 4, 9, 0.0)
CALL LRCHS7(4)
EOP = 0.0
DO 60 JS=1, NSL
IF (JS.EQ.1) EOP=1.0
ILSJ = ILS(JS)
MRLD = ITS(IS)-ILS(IS)+1
60 CALL LRGCURV(MSL(ILS, JS), WLSSL(ILS, JS), MBLD, 2, SYM, EOP)
C--FLOT PRESSURE SURFACE VELOCITIES FOR ALL STREAMLINES

CALL LRLEGN(TITL5,64,0,1.2,0.7,0.0)
CALL LRCHS7(2)
CALL LRLEGN(TITL6,24,0,3.4,1.3,0.0)
CALL LRLEGN(TITL7,8,1.0,2.4,9.0,0)
CALL LRCHS7(4)
EOP = 0.0
DC 70 JS=1,NSL
IF (JS.EQ.NSL) EOP=1.0
ILSJ = ILS(JS)
MPLD = ITS(JS)-ILS(JS)+1
70 CALL LRCURV(NSL(ILSJ,JS),WTSSL(ILSJ,JS),MBLC,2,SYM,EOP)
CALL LRCURV(NSL,RLS,0,1,SYM,1.0)
RETURN
END

SUBROUTINE TVELCY
C
C--TVELCY CALCULATES THE FULL MASSFLOW, TRANSONIC SOLUTION
C--USING VELOCITY GRADIENT EQUATIONS
C
COMMON SRW,SRE,ITER,IEND,NREAD,NWRT
COMMON /INPUTT/GAM,AR,MSFL,OMEG,REDFA,VELOC,NNEW,WNEW,WRI,WBO,
1 MM,MHT,NBL,NHIB,NTIP,NIN,NOUT,NBLPL,NPPP,NOSTAT,NSL,LSFR,
2 LTP,LAUVT,INMESH,SLINE,ISTAT,IFLC,ISUPER,ITCONS,DEBUG,
3 ZMIN,ZMID,ZMBO,ZMOUT,ZHI,ZTI,ZHOUT,HTCUT,HUP(5O),
4 RHUB(5O),ZTIP(5O),RTIP(5O),SFIN(5O),RAEGN(5O),PRCP(5O),LOESUX(5O),
5 LAMIN(5O),VTHIN(5O),SFOUT(5O),RACOUT(5O),PRCP(5O),LOESUX(5O),
6 LAMOUT(5O),VTHOUT(5O),ZHST(5O),ZSTI(5O),FLFR(5O),
7 ZBL(5O,5O),RBL(5O,5O),THRL(5O,5O),TNRL(5O,5O),
COMMON /CALCON/MMM1,MHTP1,CP,EXPCH,TGRC,PITCH,CURVHI,CURVVI,
1 CURVHO,CURVT0,RFIN,RFIN,RHOUT,RHOUT,RLEH,RLET,RTEH,RET,
2 ZLE(5O),ZLHE(5O),ZTE(5O),ZTE(5O),ZLHE(5O),ZTE(5O),
3 SLOEM(101),TLEOM(101),ZTECM(101),RTEOM(101),STECM(101),
4 THTEOM(101),ILE(101),ITE(101),ZOM(100,101),RMS(100,101),
5 SCM(100,101),TOM(100,101),BTHR(100,101),BTHC(100,101),
6 DTOM(101,101),PL0S(100,101),CPC(100,101),SFC(100,101),
COMMON /VRCM/AL4,100,101,K(100,101),K(100,101),K(100,101),
1 WSUBSI(100,101),WSUBT(100,101),WSUBT(100,101),WSUBT(100,101),
2 WSURM(100,101),WTHI(100,101),WTHI(100,101),WTHI(100,101),
3 ALPE4(100,101),RTR(100,101),RTR(100,101),RTR(100,101),
4 WLURF(100,101),WTSURF(100,101),CAMP(100,101),CAMP(100,101),
5 RMDAV(100,101),CPRV4(100,101),FR(100,101),DFCM(100,101),
6 XMOM(100,101),ZETOM(100,101),DLDU(100,101)
DIMENSION DWMDOS(100,101),DWDTS(100,101),DVT(101)
1 WMVT(101),WTVT(101),WMTK(101),CTPI(101),RCCAB(101),
2 DMVT(101),DNTVER(101),ATVEL(101),BATVEL(101),CCTVEL(101),
3 VTVEL(101),ETVEL(101),FTVEL(101),LAMBO(101),LAMBO(101),
4 TPT(101),TOP(101),RTHO(101),RHOOP(101),
5 DWDMO(100,101),DNTM(100,101),DNTM(100,101),CWTDT(100,101),
REAL MSFL,LABDA,LABBO,LAMOUT,LAMIN,LAMDAF
INTEGER SRW,SRE
LOGICAL REPEAT
C--RESTORE FULL MASS FLOW VALUES, AND REINITIALIZE LAMDAF AND RVHTA
C
IEND = IEND+1
J7 = 1
IF (REDFAC.EQ.1.0) JZ=2
IF (REDFAC.EQ.1.0) GO TO 60
WRITE(NWRT,1040)
OMEGA = OMEGA/REDFAC
MSFL = MSFL/REDFAC
DC 10 J =1,NIN
LAMIN(J) = LAMIN(J)/REDFAC
10 VTHIN(J) = VTHIN(J)/REDFAC
DC 20 J =1,NOUT
LAMOUT(J) = LAMOUT(J)/REDFAC
20 VTHOUT(J) = VTHOUT(J)/REDFAC
CALL LAMNIT
CALL RVTNIT
C
C--CALCULATE PARTIALS WITH RESPECT TO T OF WSUBM AND WSUBT
C
DC 40 I=1,MM
DC 30 J=1,MHTP1
TVERT(J) = TOM(I,J)
WMVERT(J) = WSMVM(I,J)
30 WTVERT(J) = WTH(I,J)
CALL SLOPES(TVERT,WMVERT,MHTP1,DWMVER)
CALL SLOPES(TVERT,WTVERT,MHTP1,DWTVER)
DC 40 J=1,MHTP1
DWMDT(I,J) = DWMVER(J)
40 DWTDT(I,J) = DWTVER(J)
C
C--CALCULATE PARTIALS WITH RESPECT TO S OF WSMVM AND WSMVT, AND THEN
C--CALCULATE PARTIALS WITH RESPECT TO M OF WSMVM AND WSMVT
C
DO 50 J=1,MHTP1
CALL SLOPES(SOM(I,J),WSUBM(I,J),MM,DWMDM)
CALL SLOPES(SCM(I,J),WSTM(I,J),MM,DWTCS)
DO 50 I=1,MM
DWMOS(I,J) = (DMSOS(I)*CAMP(I,J)+DWMOS(I,J)*SAMP(I,J))/REDFAC
50 DWTOS(I,J) = (DWTOS(I)*CAMP(I,J)+DWTOS(I,J)*SAMP(I,J))/REDFAC
RTOLER = 1.E-4
CHLIP = MSFL
MEAN = MHT/2+1
C
C--SOLVE VELOCITY GRADIENT EQUATION ON EACH VERTICAL MESH LINE
C
60 DO 280 I=1,MM
WHUB = WTH(I,J)/REDFAC
DELMAX = WTH(I,MEAN)/2.0/REDFAC
NCOUNT = 0
C
C--CALCULATE COEFFICIENTS A, B, AND C FOR THE VELOCITY GRADIENT EQUATION
C--INITIALIZE COEFFICIENT C TO ZERO
C
DC 80 J=1,MHTP1
LAMDAF(I,J) = LAMDAF(UOM(I,J),I,J)
LAMDD(I,J) = RVHTA(UOM(I,J),I,J)
TIPT(J) = TIPF(UOM(I,J))
TOP(J) = TOPF(UOM(I,J))
RHOIP(J) = RHOIPF(UOM(I,J))
RHOOP(J) = RHOPPF(UOM(I,J))
ATVEL(J) = 0.  
CTVEL(J) = 0.  
DTVEL(J) = 0.  
IF(I.LT.ILE(J).OR.I.GT.ILE(J)) GO TO 70  
SAL = SIN(ALPHA(I,J)).  
SBETA = SIN(BETA(I,J)).  
Cbeta = COS(BETA(I,J)).  
ATVEL(J) = CBETA**2*CAMP(I,J)*CURV(I,J)-SBETA**2*CPHI(I,J)/  
1*CM(I,J)*COTH(I,J)*SAL*CBETA*SBETA  
ATVEL(J) = CBETA*SAMP(I,J)*OAMD(I,J)-2.*OMEGA*SBETA*CPHI(I,J)/  
1+ROM(I,J)*COTH(I,J)*CBETA*(DMCD(I,J)+2.*OMEGA*SAL)  
GO TO 80  
70 ATVEL(J) = CAMP(I,J)*CURV(I,J)  
DTVEL(J) = OAMD(I,J)*SAMP(I,J)  
GO CONTINUE  
C--CALCULATE COEFFICIENT FOR THE VELOCITY GRADIENT EQUATION AND OTHER  
C--CONSTANTS FOR CHECKING CONTINUITY  
90 DO 120 J=1,MTPTP1  
OMR2 = OMEGA*ROM(I,J)**2  
TMLM(J) = 2.*OMEGA*LAMBDA(J)-CMEGA*CMR2  
CPTP(J) = 2.*CP*TIPT(J)  
IF(I.LT.ILE(J)) GO TO 100  
WHIRL = LAMBDA(J)  
TEMPER = TIPT(J)  
DENS = RHOIP(J)  
GO TO 110  
100 IF(I.LE.ILE(J)) GO TO 120  
WHIRL = LAMBDA(J)  
TEMPER = TOP(J)  
DENS = RHOIP(J)  
110 CTVEL(J) = -(WHIRL-OMR2)/ROM(I,J)**2*(CURV(I,J)*(WHIRL-OMR2)*  
1*CAMP(I,J)+(WHIRL+CMR2)/ROM(I,J)*CPHI(I,J))  
120 RCARR(J) = RHOIP(J)*CAMP(I,J)*ROM(I,J)*BTH(I,J)  
C--CALCULATE COEFFICIENTS E AND F FOR THE VELOCITY GRADIENT EQUATION  
TPP = TIPT(1)-TMLM(1)/2.*CP  
IF(TPP.LT.0.) GO TO 290  
PREL = RHOIP(1)*AR*TIPT(1)*(TPP/TIPT(1))**(GAM*EXPON)*(1.-  
1*IPLOSS(I,J))  
DC 130 J=2,MTPTP1  
DTTP = TIPT(J)-TIPT(J-1)  
DLAM = LAMBDA(J)-LAMBDA(J-1)  
TPPN = TIPT(J)-TMLM(J)/2.*CP  
IF(TPPN.LT.0.) GO TO 290  
PRELN = RHOIP(J)*AR*TIPT(J)*(TPPN/TIPT(J))**(GAM*EXPON)*(1.-  
1*IPLOSS(I,J))  
DTTP = TPPN-TPP  
DPREL = PRELN-PREL  
ETVEL(J-1) = CP*DTTP-OMEGA*DLAM-CP*DTTP+AR/(PREL+PREL)*(TPPN+TPP)  
1*CPREL  
FTVEL(J-1) = DTTP/(TPPN+TPP)-AR/CP*CPREL/(PRELN+PREL)  
TPP = TPPN  
130 PREL = PRELN  
C--OBTAIN NUMERICAL SOLUTION TO THE VELOCITY GRADIENT EQUATION  
C--FOR AN ESTIMATED VALUE OF W AT THE HUB  
C) REPEAT = .FALSE.
140 IND = 1
150 WI,J) = WHUR
   NCOUNT = NCOUNT + 1
   IF (1.GE.ILE(I) .AND. I.LE.ILE(1)) GO TO 160
   WHIRL = LAMBDA(I)
   IF (1.GT.ITE(1)) WHIRL = LAMBDA(J)
   SRETA = (WHIRL/ROM(I,J)-OMEGA*ROM(I,J))/WHUB
   IF(ABS(SRETA).GT.1.) GO TO 210
   BET(A(I,J) = ARSIN(SRETA)
160 CRETA = COS(BETA(I,J))
170 WSO = WHUB**2
   TTIP = 1.-(WSQ+TWLMPI/I)/CPTIP(I)
   IF(TTIP.LT.0.) GO TO 220
   RVA = TTIP**EXPON*WHUB*C Beta*RCARB(I)
   DO 200 J=1,MHT
   DELTA = T0M(I,J+1)-T0M(I,J)
   WAS = W(I,J)*(ATVEL(J)+PTVEL(J)+CTVEL(J)/W(I,J)+CBETA*WJVEL(J))/W(I,J)+CBETA*
   CDVEL(J)*DELTA+ETVEL(J)*W(I,J)+FTVEL(J)*W(I,J)
   IF (1.GE.ILE(J+1) .AND. I.LE.ILE(J+1)) GO TO 180
   WHIRL = LAMBDA(J+1)
   IF (1.GT.ITE(I+1)) WHIRL = LAMBDA(J)
   WTHETA = (WHIRL/ROM(I,J+1)-OMEGA*ROM(I,J+1))/WAS
   SRETA = WTHETA / WAS
   IF(ABS(SRETA).GT.1.) GO TO 210
   BET(A(I,J+1) = ARSIN(SBETA)
180 CREAT = COS(BETA(I,J+1))
   WASS = W(I,J)*(ATVEL(J+1)+PTVEL(J+1)+CTVEL(J+1)/WAS+CBETA*
   IDVEL(J+1)*DELTA+ETVEL(J+1)*WAS+CTVEL(J+1)/W(I,J)+CBETA*
   WTVEL(J+1)/WAS)
   W(I,J+1) = (WAS+WASS)/2.
   WSO = W(I,J+1)**2
   TTIP = 1.-(WSQ+TWLMPI/I+1)/CPTIP(I+1)
   IF(TTIP.LT.0.) GO TO 220
   IF(I.GE.ILE(I+1) .AND. I.LE.ILE(I+1)) GO TO 190
   SRETA = WTHETA/W(I,J+1)
   IF(ABS(SRETA).GT.1.) GO TO 210
   BET(A(I,J+1) = ARSIN(SBETA)
190 CREAT = COS(BETA(I,J+1))
   RVAS = TTIP**EXPON*W(I,J+1)*CBETA*RCARB(J+1)
   UOM(I,J+1) = (RVA+RVAS)*DELTA/2.+UCM(I,J)
200 RVA = RVAS
C
C--CHECK CONTINUITY AND ESTIMATE NEW VALUE FOR W AT THE HUB
C
   IF(IND.GE.6 .AND. ARS(MSFL-UOM(I,MHTPI)).LE.MSFL*RTOLER) GO TO 250
   CALL CONTINUE(WHUR,UOM(I,MHTPI),IND,JZ,MSFL,DELMAX)
   IF(IND.LT.10) GO TO 150
   IF(IND.EQ.10) GO TO 250
   GO TO 230
210 WHIRL = WHUP+0.5*DELMAX
   IF(NCOUNT.LT.1000) GO TO 140
   GO TO 230
220 WHIRL = WHUP-0.5*DELMAX
   IF(NCOUNT.LT.1000) GO TO 140
230 WRITE (NWRITE,1010) I
   IMESH = I
   ISTATL = 0
   DC 240 J=1,MHTPI
240 UOM(I,J) = UOM(I,J)/MSFL
   GO TO 280

175
C--SOLUTION OBTAINED, CHECK ACCURACY OF TIP, LAMBDCA, AND RHOIP

C 250 CONTINUE
DC 260 J=2,MHTPI
UOM(I,J) = UOM(I,J)/MSFL
TVAR = TIPF(UOM(I,J))
IF(ABS(TVAR-TPT(J)).GT.TVAR*RTOLER) REPEAT = .TRUE.
TPT(J) = TVAR
TVAR = TOPF(UOM(I,J))
IF(ABS(TVAR-TOP(J)).GT.TVAR*RTOLER) REPEAT = .TRUE.
TOP(J) = TVAR
TVAR = RHOIPF(UOM(I,J))
IF(ABS(TVAR-RHOIP(J)).GT.TVAR*RTOLER) REPEAT = .TRUE.
RHOIP(J) = TVAR
TVAR = LAMDAF(UOM(I,J),I,J)
IF(ABS(TVAR-LAMBOA(J)).GT.ABS(TVAR)*RTOLER) REPEAT = .TRUE.
LAMBOA(J) = TVAR
TVAR = RVHTA(UOM(I,J),I,J)
IF(ABS(TVAR-RHOOP(J)).GT.TVAR*RTOLER) REPEAT = .TRUE.
RHOOP(J) = TVAR
WHUB = WI(1,1)
IF(REPEAT.AND.NCOUNT.LT.1000) GO TO 90
IF(IND.NE.10) GO TO 270
CHFJ = UOM(I,MHTPI)*MSFL*FLCAT(NBL)
CHLIM = AMIN1(CHLIM4,CHFL)
WRITE(NWRIT,1000) CHFJ
270 IF(REPEAT) WRITE(NWRIT,1010)
280 CONTINUE

C--FINISHED VELOCITY GRADIENT SOLUTION ON EACH VERTICAL MESH LINE
C--CHECK CHOKE LIMIT
IF(CHLIM.GT.(0.9999*MSFL)) RETURN
ISUPER = 2
WRITE(NWRIT,1030) MSFL,CHLIM
RETURN
290 WRITE(NWRIT,1020)
STOP

C--FORMAT STATEMENTS
C
1000 FORMAT (69PLMSFL EXCEEDS CHOCLING MASS FLOW FOR VERTICAL ORTHOGONAL I MESH LINE I =,13/22H
CHOKING MASS FLOW =,G15.6)
1010 FORMAT (88HL A VELOCITY GRADIENT SOLUTION CANNOT BE OBTAINED FOR I VERTICAL ORTHOGONAL MESH LINE I =,13/4X,56H ANY SUBSEQUENT OUTPUT F
20R THAT MESH LINE MAY BE IN ERROR)
1020 FORMAT (62HL THE UPSTREAM INPUT WHIRL OR TANGENTIAL VELOCITY IS TOO LARGE)
1030 FORMAT (51HL CHOKING MASSFLOW IS LESS THAN THE INPUT MASSFLOW/6X,
116INPUT MASSFLOW =,G13.5/6X,26MHMINIMUM CHOKING MASSFLOW =,G13.5/26X,92HA SOLUTION CAN ONLY BE OBTAINED IF INPUT MASSFLOW IS LESS TH
3AN THIS MINIMUM CHOCLING MASSFLOW)
1040 FORMAT (/1/52X,251H*/1/52X,25H* Full MASSFLOW */42X,451H*)
1/42X,1H*/12X,19HTRANSONIC SOLUTION,12X,1H*/42X,45H* BY VELOCITY
26GRADIENT APPROXIMATE METHOD */42X,451H*)//////)
END
FUNCTION TOPF(SF)
C--TCPF calculates downstream absolute total temperature
C--as a function of stream function

COMMON/INPUT/GAM,AR,MSFL,OMEGA,REDFAC,VELTOL,FNEW,DNEW,MBI,MBO,
      MMB,MBT,NBL,NHUB,NTIP,NIN,NOUT,NALPL,NPPP,NSTAT,NSL,LSFR,
      LMPL,LMWT,IMESH,ISLINE,ISTATL,IPLOT,ISR,ITSCN,ITSCN,ITSCN
      TOTMIN,TOTMTH,TOTMCAP,TSF,TOUT,ZIN,ZHOUT,ZHOUT,ZHOUT,ZHOUT
      RHUB(50),RHI(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
      LAMIN(50),VTIN(50),SFOUT(50),RADOUT(50),PRCP(50),LOSSOUT(50),
      LAMOUT(50),VTCHT(50),ZHTST(50),ZTST(50),FLFR(50),
      ZBL(50,50),RLB(50,50),TMBL(50,50),TNBL(50,50)
COMMON/CALCON/MMMI,MMHTP,CP,EXPN,TGROG,PITCH,CURVHI,CURVTI,
      CURVHO,CURVTO,RHIN,RTH,RTCHUT,RTCHUT,RTCHUT,RTCHUT
      ZLE(50),ZLE(50),ZTE(50),ZTE(50),ZLEOM(101),ZLEOM(101),
      SLEOM(101),SLEOM(101),ZTECM(101),ZTECM(101),
      THTEOM(101),THTEOM(101),ZTEM(101),ZTEM(101),
      STHTEOM(101),STHTEOM(101),THTEOM(101),THTEOM(101),
      Z0M(100,101),ZOM(100,101),ZOM(100,101),
      ZOTHOT(100,101),ZOTHOT(100,101),ZOTHOT(100,101),
      ZOTHOT(100,101),ZOTHOT(100,101)

REAL LAMDAF
TOPF = TIPF(SF)-OMEGA/CP*(LAMDAF(SF, ILE(1), 1)-RVTHTA(SF, ILE(1), 1))
RETURN
END

FUNCTION TIPF(SF)
C--TIPF calculates upstream absolute total temperature
C--as a function of stream function

COMMON/INPUT/GAM,AR,MSFL,OMEGA,REDFAC,VELTOL,FNEW,DNEW,MBI,MBO,
      MMB,MBT,NBL,NHUB,NTIP,NIN,NOUT,NALPL,NPPP,NSTAT,NSL,LSFR,
      LMPL,LMWT,IMESH,ISLINE,ISTATL,IPLOT,ISR,ITSCN,ITSCN,ITSCN
      TOTMIN,TOTMTH,TOTMCAP,TSF,TOUT,ZIN,ZHOUT,ZHOUT,ZHOUT,ZHOUT
      RHUB(50),RHI(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
      LAMIN(50),VTIN(50),SFOUT(50),RADOUT(50),PRCP(50),LOSSOUT(50),
      LAMOUT(50),VTCHT(50),ZHTST(50),ZTST(50),FLFR(50),
      ZBL(50,50),RLB(50,50),TMBL(50,50),TNBL(50,50)
COMMON/CALCON/MMMI,MMHTP,CP,EXPN,TGROG,PITCH,CURVHI,CURVTI,
      CURVHO,CURVTO,RHIN,RTH,RTCHUT,RTCHUT,RTCHUT,RTCHUT
      ZLE(50),ZLE(50),ZTE(50),ZTE(50),ZLEOM(101),ZLEOM(101),
      SLEOM(101),SLEOM(101),ZTECM(101),ZTECM(101),
      THTEOM(101),THTEOM(101),ZTEM(101),ZTEM(101),
      STHTEOM(101),STHTEOM(101),THTEOM(101),THTEOM(101),
      Z0M(100,101),ZOM(100,101),ZOM(100,101),
      ZOTHOT(100,101),ZOTHOT(100,101),ZOTHOT(100,101),
      ZOTHOT(100,101),ZOTHOT(100,101)
DIMENSION SLOPE(50,50),SLOPE(50,50)
K = 2
IFABS(SF-SFIN(1)).GT.TOLER) GO TO 10
    TIPF = TIP(1)
RETURN
10 IF(SF=SFIN(1)) 20,20,30
    TIPF = TIP(1)+(SF-SFIN(1))*SLOPE(1)
RETURN
30 IFABS(SF-SFIN(K)).GT.TOLER) GO TO 40
    TIPF = TIP(K)
RETURN
40 IF(SF-SFIN(1)) 70,70,50
50 K=K+1
   IF(K-NIN) 30,30,60
60 TPF = TIP(NIN)*(SF-SFIN(NIN))*SLOPE(NIN)
RETURN
70 SK = SFIN(K)-SFIN(K-1)
   TPF = EM(K-1)*(SF(SF)-SF)**3/6./SK+EM(K)*(SF-SFIN(K-1))**3/
1 6./SK*(TIP (K)-SK*EM(K)*SK/6.)*(SF-SFIN(K-1)+TIP(K-1)/
2 SK-EM(K-1)*SK/6.)*(SFIN(K)-SF)
RETURN
ENTRY TIP,NIN)
CALL SPLINE(SFIN, TIP, NIN, SLOPE, EM)
TOLER = ABS(SFIN(NIN)-SFIN(1))/FLCAT(NIN)+1.E-6
RETURN
END

FUNCTION RHOIPF(SF)
C RHOIPF CALCULATES UPSTREAM ABSOLUTE TOTAL DENSITY
C AS A FUNCTION OF STREAM FUNCTION
C
COMMON/INPUT/GAM,AR,MSFL,OMEGA,REDFA,VELTOL,FNEW,ONEW,BIM,MBI,
1 MM,MHT,NBL,NHLU,NTP,NIN,NCUT,NALPL,NAPPL,ANPPL,NOST AT,NSL,LSIFR,
2 LTP1,LAMVT,IMEKH,ISLINE,LSTAT,LIPLOT,LIPUIR,IUPER,IUSCN,IDEBUG,
3 TWIN,TWNI,ZCMB0,ZOMOUT,ZHIN,ZT IN,TTOUT,ZHOUR(50),
4 RHUR(50),ZTIP(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTHIN(50),SFOUT(50),RADOUT(50),PRCP(50),LOSSOUT(50),
6 LAMOUT(50),VTHOUT(50),ZHST(50),7HST(50),FLFR(50),
7 ZRL(50,50),RRL(50,50),THBL(50,50),TN8L(50,50)
COMMON/CCON/MMM1,MHTP1,CP,EXPO,TGROW,PITCH,CURVHI,CURVTI,
1 CURVHO,CURVTO,RHIN,RTIN,RHCUT,ROUT,RELH,RELRT,REH,
2 ZL(50),RLE(50),ZTE(50),RT(50),ZLEM(101),RLEM(101),
3 SLEM(101),TLEM(101),ZTE(101),RLEM(101),STE(101),
4rtle(101),ITE(101),ZC(100,101),RZC(100,101),
5 SOM(100,101),TOM(100,101),BH(100,101),DTHO(100,101),
6 DTH(100,101),PLOSS(100,101),CPHI(100,101),SPHI(100,101)
DIMENSION SLOPE(50),EM(50),RHCIP(50)
K = 2
   IF(ABS(SF-SFIN(1)) .GT. TCHER) CC TO 10
   RHOIPF = RHOIPF(1)
RETURN
10 IF(SF-SFIN(1)) 20,20,30
20 RHOIPF = RHOIPF(1)+(SF-SFIN(1))*SLOPE(1)
RETURN
30 IF(ABS(SF-SFIN(K)) .GT. TCHER) CC TO 40
   RHOIPF = RHOIPF(K)
RETURN
40 IF(SF-SFIN(K)) 70,70,50
50 K=K+1
   IF(K-NIN) 30,30,60
60 RHOIPF = RHCIP(K)+(SF-SFIN(K))*SLOPE(NIN)
RETURN
70 SK = SFIN(K)-SFIN(K-1)
   RHOIPF = EM(K-1)*(SF(SF)-SF)**3/6./SK+EM(K)*(SF-SFIN(K-1))**3/

FUNCTION LAMDAF(SF, I, J)
C--LAMDAF CALCULATES PREWHIRL, LAMBDAS, AS A FUNCTION OF STREAM
C--FUNCTION UPSTREAM OF THE BLADE

COMMON SRW, SRE, ITER, IEND, NREAD, NWRT
COMMON/INPUT/GAM, AR, MSFL, OMEGA, REDAC, VELTOL, FNEW, DNEW, MB1, MB2,
MM, MHT, NLB, NHTB, NTIP, NN, NOUT, NLPL, NPPP, NOST AT, NNL, NSL, SFR,
LPL, LAMVT, MESH, ISLINE, ISTATL, IPOINT, SJOIN, IDEBUG,
ZMIN, ZOMB, ZWBD, ZOUT, ZFIN, ZOUT, ZHOUT, ZTOT, ZHUB(50),
RHUB(50), ZTOP(50), RTOP(50), SFIN(50), RAD(50), TIP(50), PR(50),
LAMIN(50), VTHIN(50), SFBOUT(50), PR(50), LAMOUT(50), VTHOUT(50),
7 ZHST(50), ZTST(50), FLFR(50),
ZRL(50, 50), RBL(50, 50), THBL(50, 50), TNBL(50, 50)
COMMON/CALCON/MMM1, MHTPI, CP, EXPON, TGROG, PITCH, CURVHI, CURVTi,
CURVH0, CURVT0, RHIN, RTIN, RHCUT, RTOUT, RLEH, RLET, RTEH, RTET,
ZLE(50), RLE(50), ZTE(50), RTE(50), ZLEOM(101), RLEOM(101),
SLEOM(101), THLEOM(101), ZTHLEOM(101), RTHLEOM(101),
4 THETEOM(101), ILE(101), ITE(101), ZCM(100, 101), ROM(100, 101),
5 SOM(100, 101), TOM(100, 101), BTH(100, 101), DTHDS(100, 101),
6 DTMOD(100, 101), PLOSS(100, 101), CPHI(100, 101), SPHI(100, 101)
COMMON/VARCOM/A(14, 100, 101), UC*(100, 101), K(100, 101), RHO(100, 101),
WSUBR(100, 101), WSBTR(100, 101), WSBZ(100, 101), WSBZ(100, 101),
2 WSBM(100, 101), WTB(100, 101), WTH(100, 101), W(100, 101),
3 ALPHA(100, 101), BETA(100, 101), WCR(100, 101), CURV(100, 101),
4 WLSURF(100, 101), WLSURF(100, 101), CAM(100, 101), SAM(100, 101),
5 RHOAV(100, 101), DELHHC(100, 101), FR(100, 101), DFDM(100, 101),
6 XINM(100, 101), ZETOT(100, 101), ODLU(100, 101)
DIMENSION SLOPE(50), EM(50), AAA(50), RILOM(101), UILOM(101)
REAL LAMDAF, LAMIN, LAMOUT
K = 2
IF(ABS(SF-SFIN(1)).GT.TOLER) GO TO 10
LAMDAF = LAMIN(1)
IF (I.LT.ILE(J)) ODLU(I,J)=SLCEP(1)
RETURN
10 IF(ISF-SFIN(1)) 20, 20, 30
20 LAMDAF = LAMIN(1)+(SF-SFIN(1))*SLOPE(1)
IF (I.LT.ILE(J)) ODLU(I,J)=SLCEP(1)
RETURN
20 IF(ABS(SF-SFIN(KK)).GT.TOLER) GO TO 40
LAMDAF = LAMIN(KK)
IF (I.LT.ILE(J)) ODLU(I,J)=SLCEP(KK)
RETURN
40 IF(ISF-SFIN(KK)) 70, 70, 50
50 KK=KK+1
IF(KK-NIN) 30, 30, 60
60 LAMDAF = LAMIN(NIN)+(SF-SFIN(NIN))*SLOPE(NIN)
IF (I.LT.ILE(J)) DLDU(I,J)=SLCPE(NIN)
RETURN
70 SK = SFIN(KK)-SFIN(KK-1)
LAMDAF = EM(KK-1)*(SFIN(KK)-SF)##3/6./SK+EM(KK)*{(SF-SFIN(KK-1))##3
1 /6./SK+LAMIN(KK)/SK-EM(KK)*SK/6.*{(SF-SFIN(KK-1))+(LAMIN(KK-1)
2 /SK-EM(KK-1)*SK/6.)*{SFIN(KK)-SF}
IF (I.LT.ILE(J)) DLDU(I,J)=-EM(KK-1)*(SFIN(KK)-SF)##2/2./SK+
1 EM(KK)*(SFIN(KK-1)-SF)##2/2./SK+(LAMIN(KK)-LAMIN(KK-1))/SK-
2 (EM(KK)-EM(KK-1))*SK/6.
RETURN
ENTRY LAMNIT(NNN)
IF (ITER.EQ.0) GO TO 100
II = NII
JJ = 1
DO 80 KK=1,MHTP1
DOM
DIST = FLOAT(KK-1)/FLOAT(MHT)
RILCM(KK) = RHIN+DIST*(RTIN-RHIN)
ZILOM = ZHN+DIST*(ZTN-ZHN)
80 CALL LININT(ZOM,ROM,UMM,MHTP1,100,101,1ZILOM,RILCM(KK),
1UIMO(KK),II,JJ)
IF (LSFR.EQ.0) CALL SPLINT(UILCM,RILCM,MHTP1,SFIN,NIN,RADIN,AAA)
IF (LSFR.EQ.1) CALL SPLINT(RILCM,UILCM,MHTP1,RADIN,NIN,SFIN,AAA)
IF (LSFR.EQ.1) GO TO 100
DO 90 KK=1,NIN
90 LAMIN(KK)= RADIN(KK)*VTHIN(KK)
100 CALL SPLINE(SFIN,LAMIN,NIN,SLCPE,EM)
TOLER = ABS(SFIN(NIN)-SFIN(1))/FLOAT(NIN)*1.E-6
RETURN
END

FUNCTION RHOOPF(SF)
C
C--RHOOPF CALCULATES DOWNSTREAM ABSOLUTE TOTAL DENSITY
C--AS A FUNCTION OF STREAM FUNCTION
C
COMMON/INPUT/GAM,AR,MSFL,OMEGA,RECFC,VELTCL,FNEW,DNEW,MBI,MBO,
1 MMT,NHT,NRL,NHUB,NTP,NIN,NOUT,NRLPL,NPPP,OSTAT,NSL,LSFR,
2 LTP,MAMTV,IMESH,ISL NEE,ISTATNL,PLOT,MUPER,ITSON,IDEBUG,
3 ZOMIN,ZOMBI,ZOMBO,ZOMOUT,ZHTN,ZTHIN,ZHOUT,ZTOUZ,ZHUB(50),
4 RHB(50),ZTIP(50),RTIP(50),SFIN(50),RADIN(50),TIP(50),PRIP(50),
5 LAMIN(50),VTHIN(50),SFOU(50),RACOUT(50),PRP(50),LOSOUT(50),
6 LAMOUT(50),VTHCUT(50),ZHST(50),ZTST(50),FLFR(50),
7 ZBL(50,50),RBL(50,50),THBL(50,50),TBL(50,50),THIEM(101),
8 COMMON/CALC/MMM1,MHTP1,CP,EXPON,TGROG,PITCH,CURVHI,STERT,
1 CURVHO,CURVTC,RHIN,RTIN,RHOUT,RTOUT,RLEH,RLET,RTHE,RTET,
2 ZLE(50),REL(50),ZTE(50),RTE(50),ZLECM(101),RELCM(101),
3 SLEC1M(101),TLECM(101),ZTECM(101),RTENM(101),STENM(101),
4 THTENM(101),ITE(101),ZOM(100,101),ROM(100,101),
5 SOM(100,101),TCM(100,101),THM(100,101),SOM(100,101),
6 DTH(100,101),DLOSS(100,101),SVPHI(100,101),SPHI(100,101)
DIMENSION SLOPE(50),EM(50),RHOOP(50)
K = 2
IF(ABS(SF-SFOUT(I)).GT.TOLER) GO TO 10
RHCPP = RHCPP(1)
RETURN
10 IF(SF-SFOUT(1)) 20,20,30
20 RHCPP = RHCPP(1)+(SF-SFOUT(1))*SLOPE(1)
RETURN
30 IF((ABS(SF-SFOUT(K))*.GT.TOLER) GO TO 40
RHCPP = RHCPP(K)
RETURN
40 IF(SF-SFOUT(K)) 70,70,50
50 K=K+1
IF(K=NOUT) 30,30,60
60 RHCPP = RHCPP(NOUT)+(SF-SFOUT(NOUT))*SLOPE(NOUT)
RETURN
70 SK = SFOUT(K)-SFOUT(K-1)
RHCPP = EM(K-1)*(SFOUT(K)-SF)**3/6./SF+EM(K)*(SF-SFOUT(K-1))**3
1 /6./SK+RHCPP(K)/SK-EM(K)*SF/6.)*(SF-SFOUT(K-1))*(RHCPP(K)/SK-EM(K)*SF/6.)*(SF-SFOUT(K-1)*(RHCPP(K-1)/
2 SK-EM(K-1)*SK/6.)*(SFOUT(K)-SF)
RETURN
ENTRY RHONIT(NNN)
DO 80 J=1,NOUT
80 RHCPP(J) = PROPJ(J)/AR/TOPF(SFOUT(J))
CALL SPLINE(SFOUT,RHCPP,NOUT,SLOPE,EM)
TOLER = ABS(SFOUT(NOUT)-SFOUT(1))/FLOAT(NOUT)*1.E-6
RETURN
END

FUNCTION RVTHETA(SF1, IJ)
C--RVTHETA CALCULATES R * V-THETA AS A FUNCTION OF STREAM FUNCTION
C--STREAMFUNCTION OF THE BLADE
C
COMMON SRW, SRE, ITER, IEND, NREAD, NWRT
COMMON/INPUTT/GAM, AR, MSFL, OMEGA, REDFAC, VELTOL, FNEW, DNEW, MBA, MBO,
1 MM, MHT, NHL, NHUB, NTIP, NIN, NOUT, NBLPL, NPPP, NOSTAT, NSL, LSF,
2 LTLPL, LAVTL, IMESH, ISLINE, ISTATL, IPOINT, ISUPER, ITSON, IDEBUG,
3 ZMIN, ZOMIN, ZOMB, ZOMOUT, ZHIN, ZTIN, ZHOUT, ZTCUT, ZHUB(50),
4 RHNIP(50), ZTIP(50), RTIP(50), SFIN(50), RADI(50), TIP(50), PRIP(50),
5 LAMIN(50), VTHIN(50), SFOUT(50), RACOUT(50), PREP(50), LOSOUT(50),
6 LAMOUT(50), VTHOUT(50), ZHST(50), ZTST(50), FLFR(50),
7 ZBL(50,50), RBL(50,50), THBL(50,50), TNBL(50,50),
8 COMMON/CALCON/MM1,MMHT,CP,EXPAN,TRMRG,PITCH, CURV, CMTT,
1 CURVH0, CURVT0, RHN1, RTIN, RHOUT, RTOUT, RLEH, RLET, RTHE, RTET,
2 ZLE(50), ZLE(50), ZTE(50), RTE(50), ZLEOM(101), RLEOM(101),
3 SLEOM(101), THLEOM(101), ZTECM(101), RTECM(101), STECM(101),
4 THEOM(101), ILE(101), ITET(101), ZOM(100,101), ROM(100,101),
5 SOM(100,101), TROM(100,101), RTH(100,101), OTHCS(100,101),
6 DTOM(100,101), PLOSS(100,101), CPHI(100,101), SPHI(100,101),
COMMON/VARCM/AA(100,101), UOM(100,101), K(100,101), RHOD(100,101),
1 WSUB(100,101), WSUBT(100,101), WSUBZ(100,101), WSUBR(100,101),
2 WSUBM(100,101), WTHI(100,101), WTH(100,101), WTH(101,101),
3 ALPHA(100,101), BETA(100,101), WWCR(100,101), CURV(100,101),
4 WLSURF(100,101), WTSURF(100,101), CAMP(100,101), SAMP(100,101),
5 RHOAV(100,101), CELRH(100,101), FR(100,101), DFM(100,101),
6 XDM(100,101), ZETOM(100,101), ODLU(100,101)
DIMENSION SLOPE(50), EM(50), AAA(50), ROLOM(101), UCLCM(101)
REAL LAMIN, LAMOUT

181
SUBROUTINE CONTIN(XEST,YCALC,IND,JZ,YGIV, XOEL)
C
C--CONTIN CALCULATES AN ESTIMATE OF THE RELATIVE FLOW VELOCITY
C--FOR USE IN THE VELOCITY GRADIENT EQUATION
C
DIMENSION X(3), Y(3)

182
NCALL = NCALL + 1
IF (IND .NE. 1 .AND. NCALL .GT. 100) GO TO 160
GO TO (10, 30, 40, 50, 60, 110, 150), IND

C--FIRST CALL
10 NCALL = 1
XORIG = XEST
IF (YCALC .GT. YGIV .AND. JZ .EQ. 1) GO TO 20
IND = 2
Y(1) = YCALC
X(1) = 0.
XEST = XEST + XDEL
RETURN
20 IND = 3
Y(3) = YCALC
X(3) = 0.
XEST = XEST - XDEL
RETURN

C--SECOND CALL
30 IND = 4
Y(2) = YCALC
X(2) = XEST - XORIG
XEST = XEST + XDEL
RETURN
40 IND = 5
Y(2) = YCALC
X(2) = XEST - XORIG
XEST = XEST - XDEL
RETURN

C--THIRD OR LATER CALL - FIND SUBSONIC OR SUPERSONIC SOLUTION
50 Y(3) = YCALC
X(3) = XEST - XCRIG
GO TO 70
60 Y(1) = YCALC
X(1) = XEST - XORIG
70 IF (YGIV .LT. AMIN(Y(1), Y(2), Y(3))) GO TO (120, 130), JZ
80 IND = 6
CALL PARC(X, Y, APA, BPR, CPC)
DISCR = BPR**2 - 4.*APA*(CPC - YGIV)
IF (DISCR .LT. 0.) GO TO 140
IF (ABS(400.*APA*(CPC - YGIV)) .LE. BPR**2) GO TO 90
XEST = -BPR - SQRT(DISCR), APA
IF (JZ .EQ. 1 .AND. APA .GT. 0. .AND. Y(3) .GT. Y(1)) XEST = -BPR +
1SORTDISCR
IF (JZ .EQ. 2 .AND. APA .LT. 0.) XEST = -BPR - SQRT(DISCR)
XEST = XEST/2./APA
GO TO 100
90 IF (JZ .EQ. 2 .AND. BPR .GT. 0.) GO TO 130
ACR2 = APA/BPR*(CPC - YGIV)/BPR
IF (ABS(ACR2) .LE. 1 - E - 8) ACR2 = 0.
XEST = -(CPC - YGIV)/BPR*(1 + ACR2 + 2.*ACR2**2)
100 IF (XEST .LT. X(1)) GO TO 120
XEST = XEST - XORIG
RETURN

C--FOURTH OR LATER CALL - NOT CHOKE
110 IF(XEST - XORIG .GT. X(3)) GO TO 130
IF(XEST - XORIG .LT. X(1)) GO TO 120
Y(2) = YCALC
X(2) = XEST - XCRIG
GO TO 70
C--THIRD OR LATER CALL - SOLUTION EXISTS.
C--PUT RIGHT OR LEFT SHIFT REQUIRED
120 IND = 5
C--LEFT SHIFT
    XEST = X(1)-XDEL+XCRIG
    XOSHFT = XEST-XORIG
    XCRIG = XEST
    Y(3) = Y(2)
    X(3) = X(2)-XOSHFT
    Y(2) = Y(1)
    X(2) = X(1)-XOSHFT
    RETURN
130 IND = 4
C--RIGHT SHIFT
    XEST = X(3)+XDEL+XORIG
    XCSHFT = XEST-XORIG
    XORIG = XEST
    Y(1) = Y(2)
    X(1) = X(2)-XOSHFT
    Y(2) = Y(3)
    X(2) = X(3)-XOSHFT
    RETURN
C--THIRD OR LATER CALL - APPEARS TO BE CHECKED
140 XEST = -BPS/2./APA
    IND = 7
    IF (XEST.LT.X(1)) GO TO 120
    IF(XEST.GT.X(3)) GO TO 130
    XEST = XEST+XOPRIG
    RETURN
C--FOURTH OR LATER CALL - PROBABLY CHECKED
150 IF (YCALC.GE.YGIV) GO TO 110
    IND = 10
    RETURN
C--NO SOLUTION FOUND IN 100 ITERATIONS
160 IND = 11
    RETURN
END

SUBROUTINE PABC(X,Y,A,B,C)
C
C--PABC CALCULATES COEFFICIENTS A,B,C OF THE PARABOLA
C--Y=A*X**2+B*X+C, PASSING THROUGH THE GIVEN X,Y POINTS
C
DIMENSION X(3),Y(3)
C1 = X(3)-X(1)
C2 = (Y(2)-Y(1))/(X(2)-X(1))
A = (C1*C2-Y(3)+Y(1))/C1/(X(2)-X(3))
B = C2-(X(1)+X(2))*A
C = Y(1)-X(1)*A-X(1)**2*A
RETURN
END
SUBROUTINE INRSCT(XCURV1,YCURV1,N1, XCURV2,YCURV2,N2, XCROSS,YCROSS)
C--INRSCT CALCULATES THE COORDINATES (XCROSS,YCROSS) OF THE POINT
C--OF INTERSECTION OF TWO SPLINE CURVES, YCURV1=F(XCURV1) AND
C--XCURV2=G(YCURV2), LYING ON A PLANE
C
COMMON SRW,SRE,ITER, IEND, NREAD,NWRIT
DIMENSION XCURV1(N1), YCURV1(N1), XCURV2(N2), YCURV2(N2)
NCOUNT = 0
TOLER = (ABS(XCURV1(1)-XCURV1(1))+ABS(YCURV2(N2)-YCURV2(1)))/1.E5
XTEMP = XCURV1(1)
YTEMP = YCURV1(1)
XCROSS = (XCURV1(1)+XCURV1(N1))/2.

C--COMPUTE INTERSECTION POINT AND SLOPE ON CURVE 1
10 X1 = XCROSS
   CALL SPLINT(XCURV1,YCURV1,N1,X1,1,Y1,S1)
   C--COMPUTE INTERSECTION POINT AND SLOPE ON CURVE 2
   Y2 = Y1
   CALL SPLINT(YCURV2,XCURV2,N2,Y2,1,X2,S2)
   C--COMPUTE COORDINATES OF POINT WHERE TWO SLOPES INTERSECT
   S1S2 = S1*S2
   XCROSS = X2+S1S2*(X2-X1)/(1.-S1S2)
   YCROSS = Y1+S1S2*(Y2-Y1)/(1.-S1S2)

C--COMPUTE DISTANCE AWAY FROM PREVIOUS SLOPE INTERSECTION POINT
   DIST = SQRT((YCROSS-YTEMP)**2+(XCROSS-XTEMP)**2)
   IF (DIST.LT.TOLER) RETURN
   NCOUNT = NCOUNT+1
   IF (NCOUNT.GT.20) GO TO 20
   XTEMP = XCROSS
   YTEMP = YCROSS
   GO TO 10
20 WRITE(NWRIT,1000) TOLER,DIST
RETURN
1000 FORMAT (6X,46HINRSCT HAS FAILED TO CONVERGE IN 20 ITERATIONS/
110X,11HTOLERANCE =,G14.6/10X,471DISTANCE BETWEEN LAST TWO INTERSEC
2TION POINTS =,G14.6)
END

SUBROUTINE ROOT(A,B,Y,FUNCT,TCLERY,X,DFX)
C--ROOT FINDS A ROOT FOR (FUNCT MINUS Y) IN THE INTERVAL (A,B)
C
COMMON SRW,SRE,ITER, END, NREAD, NWRIT
INTEGER SRW, SRE

10 IF (SRW.EQ.21) WRITE(NWRIT,1010) A,B,Y,TCLERY
    X1 = A
    CALL FUNCT(X1,FX1,DFX)
    IF (SRW.EQ.21) WRITE(NWRIT,1020) X1,FX1,DFX
    X2 = B
20 DO 40 I=1,20
    X = (X1+X2)/2.
    CALL FUNCT(X,FX,DFX)
    IF (FXI-Y)*(FX-Y).GT.0. GO TO 30
40 CONTINUE
RETURN
1010 FORMAT (6X,46HROOT HAS FAILED TO CONVERGE IN 20 ITERATIONS/
110X,11HTOLERANCE =,G14.6/10X,471DISTANCE BETWEEN LAST TWO INTERSEC
2TION POINTS =,G14.6)
END
SUBROUTINE LININT(X,Y,Z,NX, NY, NOIMX, NDIMY, XO, YO, ZO, I, J)

C--LININT LOCATES THE POINT (XO,YO) IN A 2-C MESH WITH
C--COORDINATES STORED IN THE X AND Y ARRAYS. THEN THE VALUE OF ZO AT
C--(XO,YO) IS INTERPOLATED FROM THE Z ARRAY VALUES CORRESPONDING
C--TO THE X AND Y ARRAYS

DIMENSION X(NOIMX,NDIMY), Y(NOIMX,NDIMY), Z(NOIMX,NDIMY)
DIMENSION EXTRAP(2)
INTEGER ABOVE, RIGHT
C--FIND I, J SUCH THAT (XO,YO) IS IN COLUMN I FROM THE LEFT AND IN ROW J
C--FROM THE BOTTOM

IF(NX.LT.2.OR.NY.LT.2) STOP
IF(I.LE.0) I = 1
IF(I.GE.NX) I = NX-1
IF(J.LE.0) J = 1
IF(J.GE.NY) J = NY-1

10 ABOVE = -1
RIGHT = -1
IF(YO.GE.Y(I,J)+(XO-X(I,J))/(X(I+1,J)-X(I,J))*(Y(I+1,J)-Y(I,J)))
1 ABOVE = ABOVE+1
IF(YO.GE.Y(I,J+1)+(XO-X(I,J+1))/(X(I+1,J+1)-X(I,J+1))
1 (Y(I+1,J+1)-Y(I,J+1))) ABOVE = ABOVE+1
IF(XO.GE.X(I,J)+(YO-Y(I,J))/(Y(I,J+1)-Y(I,J))*(X(I,J+1)-X(I,J)))
1 RIGHT = RIGHT+1
IF(XO.GE.X(I+1,J)+(YO-Y(I+1,J))/(Y(I+1,J+1)-Y(I+1,J))*(X(I+1,J+1)-X(I+1,J)))
1 (X(I+1,J+1)-X(I+1,J))) RIGHT = RIGHT+1
IN = I+RIGHT
JN = J+ABOVE
IF(IN.LT.1.OR.IN.GE.NX) RIGHT = 0
IF(JN.LT.1.OR.JN.GE.NY) ABOVE = 0
IF((ABOVE**2+RIGHT**2)**.5.EQ.0) GO TO 20
I = I+RIGHT
J = J+ABOVE
GO TO 10
20 IJEX = 1
C-- SET EXTRAP TO INDICATE EXTRAPOLATION
    EXTRAP(1) = 0.
    EXTRAP(2) = 0.
    IF(IN.LT.1) EXTRAP(2) = -1.
    IF(IN.GE.NX) EXTRAP(2) = 1.
    IF(JN.LT.1) EXTRAP(1) = -1.
    IF(JN.GE.NY) EXTRAP(1) = 1.
C--CALCULATE CONSTANTS TO CALCULATE FY
    Y13 = Y(I+1,J)-Y(I,J+1)
    X13 = X(I+1,J)-X(I,J+1)
    Y42 = Y(I,J+1)-Y(I+1,J)
    X42 = X(I,J+1)-X(I+1,J)
    Y01 = YO-Y(I,J)
    X01 = XO-X(I,J)
    Y02 = YO-Y(I+1,J)
    X02 = XO-X(I+1,J)
    Y21 = Y(I+1,J)-Y(I,J)
    X21 = X(I+1,J)-X(I,J)
C--CALCULATE COEFFICIENTS OF QUADRATIC EQUATION FOR FRACTIONAL DISTANCE
C-- IN QUADRILATERAL
30 QA = Y13*X42-X13*Y42
    QB = X13*Y02-Y13*X02+Y01*X42-X01*Y42
    QC = Y01*X21-X01*Y21
    DISCR = QB**2-4.*QA*QC
    IF(DISCR.LT.0.) GO TO 110
C-- CHECK TO SEE IF QUADRATIC EQUATION IS CLOSE TO LINEAR
    IF(ABS(QA*QC).LE.QB**2*.01) GO TO 80
    FA = -(QB/2.)/QA
    FB = SQRT(DISCR)/2./QA
    F1 = FA+FB
    F2 = FA-FB
C-- CHECK TO DETERMINE WHETHER F1 OR F2 IS THE PROPER SOLUTION
    CASE = -1.
    IF(EXTRAP(IJEX)) CASE = CASE+1.
    IF(F2.LT.-.99) CASE = CASE+2.
    IF(CASE.LT.1.5) GO TO 70
    CASE = CASE-1.
    IF(F1.LT.F2) CASE = CASE-1.
    GO TO 70
C-- NC EXTRAPOATION
50 IF(ABS(F1-.5).LT..5) CASE = CASE+1.
    IF(ABS(F2-.5).LT..5) CASE = CASE+2.
    GO TO 70
C-- EXTRAPOLATION ABOVE OR TO RIGHT (FF GREATER THAN 1.)
60 IF(F1.GT..99) CASE = CASE+1.
    IF(F2.GT..99) CASE = CASE+2.
    IF(CASE.LT.1.5) GO TO 70
    CASE = CASE-1.
    IF(F1.LT.F2) CASE = CASE-1.
70 IF(ABS(CASE-.5).LT..6) GO TO 110
    FF = (1.-CASE)*F1+CASE*F2
    GO TO 90
C-- IF QUADRATIC EQUATION IS NEAR LINEAR, USE BINOMIAL EXPANSION FOR FF
80 ACR2 = QA/QB*QC/QB
    IF(ABS(ACR2).LT.1.E-8) ACR2 = 0.
FF = -OC/QR*(1.*ACB2+2.*ACB2**2)

90 IF(IJEX.EQ.2) GO TO 100
IJEX = IJEX+1
FY = FF
C--INTERCHANGE CORNER POINTS TO GET FX
Y13 = Y(I+1,J)-Y(I,J)
X13 = X(I,J)-X(I+1,J)
Y42 = Y(I,J+1)-Y(I+1,J)
X42 = X(I,J+1)-X(I,J)
Y02 = V0-Y(I,J+1)
X02 = X0-X(I,J+1)
Y21 = Y(I,J+1)-Y(I,J)
X21 = X(I,J+1)-X(I,J)
GO TO 30
C--CALCULATE INTERPOLATED VALUE
100 FX = FF
70 = Z(I,J)*(1.-FX)*(1.-FY)+Z(I+1,J)*(1.-FY)*Z(I,J+1)*(1.-FX)
1. *FY+Z(I+1,J+1)*FX*FY
RETURN
C--PRINT ERROR MESSAGE IF THERE IS A PROBLEM IN OBTAINING A SOLUTION
110 Z0 = 0.
WRITE(6,1000) I,J
RETURN
1000 FORMAT(38HILININT CANNOT FIND INTERPOLATED VALUE/4H I =,16,4H J =,116)
END

SUBROUTINE SPLINE (X,Y,NgSLOPE,EM)

C--SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C--END CONDITION - SECOND DERIVATIVES AT EITHER END POINT IS
C--ONE HALF THAT AT THE ADJACENT POINT

COMMON SRW,SRE,ITER,IEND,NREAD,NWRIT
DIMENSION X(N),Y(N),EM(N),SLOPE(N)
DIMENSION G101),SB(101)
INTEGER SRW,SRE
SB(1) = -0.5
G(1) = 0.
NC=N-1
IF (NO.LT.2) SB(1)=0.
IF(NO.LT.2) GO TO 20
DC 10 I=2,NC
A = (X(I)-X(I-1))/6.
C = (X(I+1)-X(I))/6.
W = 2.*(A+C)-A*SB(I-1)
SB(I) = C/W
F = (Y(I+1)-Y(I))/*(X(I+1)-X(I))-(Y(I)-Y(I-1))/*(X(I)-X(I-1))
10 G(I) = (F-A*G(I-1))/W
20 EM(N) = G(N-1)/(2.+SB(N-1))
DO 30 I=2,N
K = N+1-I
30 EM(K) = G(K)-SB(K)*EM(K+1)
SLOPE(1) = (X1-X2)/6.*T1*EM(1)+EM(2)+Y(2)-Y(1))/(X(2)-X(1))
DO 40 I=2,N
40 SLOPE(I) = (X(I)-X(I-1))/6.*T1*EM(I)+EM(I-1)+Y(I)-Y(I-1))
SUBROUTINE SPLINT (X, Y, N, Z, MAX, YINT, CYDX)

C--SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
C--FOR A SPLINE CURVE
C--END CONDITION - SECOND DERIVATIVE AT EITHER END POINT IS ONE-HALF
C--THAT AT THE ADJACENT POINT
C
COMMON SRW, SRE, ITER, IEND, NREAD, NWRIT
DIMENSION X(N), Y(N), Z(MAX), YINT(MAX), CYDX(MAX)
DIMENSION G(101), SB(101), EM(101)
INTEGER SRW, SRE
IF (MAX.LE.0) RETURN
TOLER = ABS(X(N)-X(1))/FLOAT(N)*1.E-5
SP(I) = -.5
G(1) = 0.
NC = N-1
IF (NO.LT.2) GO TO 20
DO 10 I=2, NC
A = (X(I)-X(I-1))/6.
C = (X(I+1)-X(I))/6.
W = 2.*A*C-A*SB(I-1)
SR(I) = C/W
F = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
10 G(I) = (F-A*G(I-1))/W
20 EM(N) = G(N-1)/(Z+SB(N-1))
DO 30 I=2, N
K = N+1-I
30 EM(K) = G(K)-SB(K)+EM(K+1)
C
ENTRY SPLINT (X, Y, N, Z, MAX, YINT, CYDX)
DO 120 I=1, MAX
K = 2
IF (ABS(Z(I)-X(I)).LT.TCLER) GO TO 40
IF (Z(I).GT.X(I)) GO TO 50
GO TO 80
40 YINT(I) = Y(I)
SK = X(K)-X(K-1)
GO TO 110
50 IF (ABS(Z(I)-X(K)).LT.TCLER) GO TO 60
IF (Z(I).GT.X(K)) GO TO 70
GO TO 100
60 YINT(I) = Y(K)
SK = X(K)-X(K-1)
GO TO 110
70 K = K+1
IF (K.NE.50) GO TO 50
80 CYDX(I) = (X(I)-X(2))/6.*(Z+EM(I)+EM(2)+(Y(2)-Y(1))/(X(2)-X(1))
YINT(I) = Y(I)+CYDX(I)*(Z(I)-X(1))

189
SLOPES(X,Y,N,SLOPE)

C--SLOPES CALCULATES FIRST DERIVATIVES, SLOPE, OF THE FUNCTION, Y,
C--WITH RESPECT TO X, USING A PARABOLIC FIT THROUGH EACH SET OF
C--THREE ADJACENT POINTS ON THE CURVE

DIMENSION X(N),Y(N),SLOPE(N)
N1 = N-1
N2 = N-2
IF (N1.LT.2) GO TO 20

C--MID POINTS
DO 10 I=2,N1
X3X2 = X(I+1)-X(I)
X2X1 = X(I)-X(I-1)
X3X1 = X(I+1)-X(I-1)
Y3Y2 = Y(I+1)-Y(I)
Y2Y1 = Y(I)-Y(I-1)
10 SLOPE(I) = (X2X1**2*Y3Y2+X3X2**2*Y2Y1)/(X3X2*X2X1*X3X1)

C--FIRST POINT
X3X2 = X(3)-X(2)
X2X1 = X(2)-X(1)
X3X1 = X(3)-X(1)
Y3Y1 = Y(3)-Y(1)
Y2Y1 = Y(2)-Y(1)
SLOPE(I) = (X3X1**2*Y2Y1-X2X1**2*Y3Y1)/(X3X2*X2X1*X3X1)

C--LAST POINT
X3X2 = X(N)-X(N1)
X2X1 = X(N1)-X(N2)
X3X1 = X(N)-X(N2)
Y3Y2 = Y(N)-Y(N1)
Y3Y1 = Y(N)-Y(N2)
SLOPE(N) = (X3X1**2*Y3Y2-X3X2**2*Y3Y1)/(X3X2*X2X1*X3X1)
RETURN
C--TWO POINT FUNCTION
20 SLOPE(1) = (Y(2)-Y(1))/(X(2)-X(1))
SLOPE(2) = SLOPE(1)
RETURN
END

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, August 20, 1973,
APPENDIX A

FINITE-DIFFERENCE FORM OF STREAM-FUNCTION EQUATION

The stream-function equation was derived as equation (B18) of part I (ref. 6):

\[
\frac{\partial^2 u}{\partial s^2} + \frac{\partial^2 u}{\partial t^2} - \frac{\partial u}{\partial s} \left[ \sin \varphi + \frac{1}{r} \frac{\partial B}{\partial s} + \frac{1}{\rho} \frac{\partial \rho}{\partial s} \frac{\partial \sin \varphi}{\partial t} \right] - \frac{\partial u}{\partial t} \left[ \cos \varphi + \frac{1}{r} \frac{\partial B}{\partial t} + \frac{1}{\rho} \frac{\partial \rho}{\partial t} \cos \varphi \right] \\
\times \frac{\partial (\sin \varphi)}{\partial s} + \frac{r B \rho}{\omega \psi} \left\{ \frac{W}{\psi} \left[ \sin \varphi \frac{\partial (r V_\theta)}{\partial s} + \cos \varphi \frac{\partial (r V_\theta)}{\partial t} \right] + \xi W^2 + \zeta + F_r \right\} = 0 \quad (A1)
\]

where

\[
\xi = \frac{1}{2c_p} \left( \frac{R}{p''} \frac{\partial p''}{\partial r} - \frac{1}{T''} \frac{\partial T''}{\partial r} - \frac{\omega^2 r}{T''} \right) \quad (A2)
\]

\[
\zeta = \omega^2 r - \frac{R T''}{p''} \frac{\partial p''}{\partial r} \quad (A3)
\]

\[
F_r = - \frac{\partial \theta}{\partial r} \frac{\partial W}{\partial \theta} \quad (A4)
\]

Equation (A4) was derived as equation (B23) of part I.

The s- and t-coordinates are the coordinates along the orthogonal mesh generated by the program. At each point of this mesh where the value of the stream function is unknown, a finite-difference approximation to equation (A1) can be written. Adjacent to the boundary the boundary conditions are included. If there are n unknown values, n nonlinear equations are obtained in n unknowns. The equations are nonlinear since the coefficients involve the density, which depends on the solution; and since the final term depends on the solution in a nonlinear manner. The equations may be solved by an iterative procedure, with two levels of iteration. The inner iteration solves a linearized equation, and the outer iteration makes corrections to the linearized equation so that the solution converges to the solution of the original nonlinear equation.

A typical mesh point with the numbering used to indicate neighboring mesh points is shown in figure 24. The value of the stream function or the other variables at 0 is de-
noted by using the subscript 0, and similarly for the neighboring points. It can be shown that equation (A1) can be approximated by

\[
\begin{align*}
\left[ \frac{2u_1}{h_1(h_1 + h_2)} + \frac{2u_2}{h_2(h_1 + h_2)} - \frac{2u_0}{h_1h_2} \right] + \left[ \frac{2u_3}{h_3(h_3 + h_4)} + \frac{2u_4}{h_4(h_3 + h_4)} - \frac{2u_0}{h_3h_4} \right] \\
- \frac{u_4 - u_3}{h_3 + h_4} \left[ \sin \varphi_0 + \frac{1}{B_0} \left( \frac{B_4 - B_3}{h_3 + h_4} \right) + \frac{1}{\rho_0} \left( \frac{\rho_4 - \rho_3}{h_3 + h_4} \right) - \frac{1}{\cos \varphi_0} \left( \frac{\sin \varphi_2 - \sin \varphi_1}{h_1 + h_2} \right) \right] \\
- \frac{u_2 - u_1}{h_1 + h_2} \left[ \cos \varphi_0 + \frac{1}{B_0} \left( \frac{B_2 - B_1}{h_1 + h_2} \right) + \frac{1}{\rho_0} \left( \frac{\rho_2 - \rho_1}{h_1 + h_2} \right) + \frac{1}{\cos \varphi_0} \left( \frac{\sin \varphi_4 - \sin \varphi_3}{h_3 + h_4} \right) \right] \\
+ \frac{r_0 B_0 \rho_0}{\omega W_0} \left[ \left( \frac{W_\theta}{\omega W_0} \right)_0 \left[ \frac{\sin \varphi}{\partial t} \cos \varphi \frac{\partial (rV_\theta)}{\partial s} \right]_0 + \xi_0 W_0^2 + \xi_0 + (F_r)_0 \right] = 0 \quad (A5)
\end{align*}
\]

where the partials of $rV_\theta$ are calculated by different methods upstream, downstream, and within the blade. Upstream and downstream of the blade, equations (B19) and (B20)
of part I are used. Within the blade, a finite-difference approximation is used with values of $V_\theta$ from the previous iteration. The final result to be used in equation (A5) is

$$
\left[ \sin \varphi \frac{\partial (rV_\theta)}{\partial s} + \cos \varphi \frac{\partial (rV_\theta)}{\partial t} \right]_0 = \begin{cases} 
\frac{r_0 B_0 \rho_0 (W_z)_0}{w} \left( \frac{d\lambda}{d\varphi} \right)_{u_0} & \text{upstream} \\
\sin \varphi \frac{r_4 (V_\theta)_4 - r_3 (V_\theta)_3}{h_3 + h_4} + \cos \varphi \frac{r_2 (V_\theta)_2 - r_1 (V_\theta)_1}{h_1 + h_2} & \text{within blade} \\
\frac{r_0 B_0 \rho_0 (W_z)_0}{w} \left[ \frac{d(rV_\theta)}{du} \right]_0 & \text{downstream}
\end{cases}
$$

(A6)

In setting up the equations for solution, the coefficients of the $u_1$ in equation (A5) must be calculated. This was done by expressing equation (A5) as

$$
u_0 = \sum_{i=1}^{4} a_i u_1 + k_0
$$

(A7)
where

\[ a_0 = \frac{2}{h_1 h_2} + \frac{2}{h_3 h_4} \]

\[ c_1 = h_1 + h_2 \]

\[ c_2 = h_3 + h_4 \]

\[
\begin{align*}
  d_1 &= \frac{B_2 - B_1}{B_0} + \frac{\rho_2 - \rho_1}{\rho_0} + \frac{\cos \varphi_0}{c_1} + \frac{\sin \varphi_4 - \sin \varphi_3}{c_2 \cos \varphi_0} \\
  d_2 &= \frac{B_4 - B_3}{B_0} + \frac{\rho_4 - \rho_3}{\rho_0} + \frac{\sin \varphi_0}{c_2} - \frac{\sin \varphi_2 - \sin \varphi_1}{c_1 \cos \varphi_0}
\end{align*}
\]

\[ a_1 = \frac{\left( \frac{2}{h_1} + d_1 \right)}{a_0 c_1} \]

\[ a_2 = \frac{\left( \frac{2}{h_2} - d_1 \right)}{a_0 c_1} \]

\[ a_3 = \frac{\left( \frac{2}{h_3} + d_2 \right)}{a_0 c_2} \]

\[ a_4 = \frac{\left( \frac{2}{h_4} - d_2 \right)}{a_0 c_2} \]
Equation (A8) is written in the form corresponding to the calculation of the coefficients in subroutine COEF. The constant $k_0$ is calculated from equation (A9) in subroutine COEF. The quantities $\xi$ and $\zeta$ are calculated in subroutine NEWRHO from equations (A2) and (A3). The quantity $F_r$ is calculated in subroutine BLDVEL when the blade surface velocities are calculated. The quantities $d\lambda/du$ and $d(rV_\theta)/du$ are calculated by subroutines LAMDAF and RVTHTA when they are called by NEWRHO to calculate $\lambda$ or $(rV_\theta)$.
APPENDIX B

MATCHING UPSTREAM AND DOWNSTREAM FLOW CONDITIONS

TO STREAM-FUNCTION SOLUTION

The work done by each blade row is determined by the change in whirl along streamlines. That is,

\[ H_0 - H_1 = \omega \left[ (rV_\theta)_0 - \lambda \right] \]  \hspace{1cm} (B1)

In this program, whirl can vary as desired from hub to tip, but for each streamline the work done is determined by equation (B1). Also, the equation relating velocity \( W \) to temperature and density requires knowledge of upstream total temperature and whirl for that particular streamline. For this reason, it is most desirable to express upstream and downstream conditions as a function of stream function rather than radius. However, if experimental data are being used, measurements are obtained as a function of position or radius. In this case the stream function is not known, but the distribution by radius can be used for input to the program. Then by estimation and iteration the correct distribution by stream function will be obtained.

If whirl is given as a function of stream function as input (i.e., \( LSFR = LAMVT = 0 \)), no changes need be made after the first initialization. If tangential velocity \( V_\theta \) is given as input (\( LAMVT = 1 \)), certain subroutines must be reinitialized in every iteration. There are two possibilities: one that \( V_\theta \) is given as a function of stream function (\( LSFR = 0 \)), and the second that \( V_\theta \) is given as a function of radius (\( LSFR = 1 \)). In either case, what is needed is the relation between stream function and radius along the input lines. This relation is determined by the stream-function solution obtained by SOR. In each iteration, then, reinitialization calls are made by NEWRHO, if \( LAMVT = 1 \). If \( LSFR = 0 \), SFIN and SFOUT are given as input, and RADIN and RADOUT are corrected by the initialization calls to LAMNIT and RVTNIT. If \( LSFR = 1 \), RADIN and RADOUT are given as input, and SFIN and SFOUT are corrected by the same calls. In either case, SPLINT calls are made to readjust the spline fit coefficients for all five subroutines, LAMDAF, RVTHTA, TIPF, RHOIPF, and RHOOPF.
APPENDIX C

CALCULATION OF PARTIAL DERIVATIVES OF THETA

ON ORTHOGONAL MESH

In the THETOM subroutine, $\partial \theta / \partial s$ and $\partial \theta / \partial t$ are calculated at the orthogonal mesh points which lie between the leading and trailing edges of the blade. This process is executed in a series of intermediate steps because input points on the different blade planes are not required to fall on a smooth curve running from hub to shroud. Also, the angle $\phi$ is known only at mesh points, so that $\partial \theta / \partial s$ and $\partial \theta / \partial t$ cannot be obtained directly. Therefore, $\partial \theta / \partial z$ and $\partial \theta / \partial r$ are obtained as an intermediate step. It is more accurate to calculate partial derivatives first and then interpolate and transform the partials than it would be to interpolate $\theta$ itself and then calculate the partials along mesh lines.

The orthogonal mesh on a typical blade is illustrated in figure 25. Notice that some of the $t$ mesh lines cross the leading and trailing edges of the blade. To alleviate the problem of finding $\theta$-gradients on this mesh, they are first obtained on an alternative mesh, shown in figure 26, of $s'$- and $t'$-coordinates which lie entirely within the blade. Then by interpolation they are obtained at the desired orthogonal mesh points. (Recall that $z$, $r$, and $\theta$ are given as input on a number of blade sections from hub to shroud.)

The step-by-step procedure to obtain $\partial \theta / \partial s$ and $\partial \theta / \partial t$ is as follows:

1. Calculate $z$- and $r$-coordinates (ZPC and RPC) of the mesh points on the $s'$-$t'$ mesh. The $s'$ mesh lines lie along the input blade sections (ZBL,RBL points). The $t'$ mesh lines run from hub to shroud at 10-percent meridional chord locations (fig. 26). Once $z$-coordinates are calculated along the input blade sections, SPLINT calls are
made in the $s'$-direction to obtain the corresponding $r$-coordinates and angles of the $s'$-coordinate line with respect to the $z$-axis $\alpha_{s'}$. (See fig. 27.)

Figure 27. - Relation of $s'$-$t'$ mesh to $z$- and $r$-directions.

(2) Calculate arc length $SZRBL$ along the input blade section ($s'$-coordinate line) by using the input $ZBL,RBL$ coordinates.

(3) Calculate arc length $SZRPC$ along the $s'$-coordinate line by using the calculated $ZPC,RPC$ coordinates.

(4) By using SPLINT calls in the $s'$-direction, calculate $\theta$ and $\partial \theta / \partial s'$ at the $ZPC,RPC$ points.

(5) By using SPLINE calls along the $t'$-coordinate line, calculate angles between the $t'$ lines and the radial direction $\alpha_{t'}$. (See fig. 27 for sign of $\alpha_{t'}$.)

(6) Calculate arc length $SZRPC$ along the $t'$-coordinate line by using the $ZPC,RPC$ coordinates.
(7) By using SPLINE calls in the t'-direction, calculate $\partial \theta / \partial t'$ at the ZPC, RPC points.

(8) Calculate $\partial \theta / \partial z$ and $\partial \theta / \partial r$ from $\partial \theta / \partial s'$ and $\partial \theta / \partial t'$ at the $s'$- and $t'$-coordinate points, with the following equations:

$$\frac{\partial \theta}{\partial z} = \frac{\partial \theta}{\partial s'} \frac{\cos \alpha_{t'}}{\cos(\alpha_{s'} + \alpha_{t'})} - \frac{\partial \theta}{\partial t'} \frac{\sin \alpha_{s'}}{\cos(\alpha_{s'} + \alpha_{t'})}$$  \hspace{1cm} (C1)

$$\frac{\partial \theta}{\partial r} = -\frac{\partial \theta}{\partial s'} \frac{\sin \alpha_{t'}}{\cos(\alpha_{s'} + \alpha_{t'})} + \frac{\partial \theta}{\partial t'} \frac{\cos \alpha_{s'}}{\cos(\alpha_{s'} + \alpha_{t'})}$$  \hspace{1cm} (C2)

(The $\partial \theta / \partial z$ and $\partial \theta / \partial r$ gradients are the ones which will be interpolated back to the orthogonal mesh and then transformed to get $\partial \theta / \partial s$ and $\partial \theta / \partial t$.)

(9) Interpolate, by using LININT calls, from $\partial \theta / \partial z$ and $\partial \theta / \partial r$ on the $s'$-t' mesh to obtain $\partial \theta / \partial z$ and $\partial \theta / \partial r$ at the $s$-t points of the orthogonal mesh which lie between the leading and trailing edges of the blade.

(10) Rotate the r- and z-coordinate lines through the angle $\varphi$ to obtain $\partial \theta / \partial s$ and $\partial \theta / \partial t$ at the orthogonal mesh points within the blade (see fig. 28). The following equations are used:

$$\frac{\partial \theta}{\partial s} = \frac{\partial \theta}{\partial z} \cos \varphi + \frac{\partial \theta}{\partial r} \sin \varphi$$  \hspace{1cm} (C3)

$$\frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial r} \cos \varphi - \frac{\partial \theta}{\partial z} \sin \varphi$$  \hspace{1cm} (C4)

Figure 28. - Relation of z- and r-directions to s- and t-directions.
APPENDIX D

LINEAR INTERPOLATION IN A QUADRILATERAL

There are several instances where it is required for the program to interpolate from a two-dimensional array of values on a grid. If the grid were rectangular, this would be straightforward. However, usually this is not the case. In most cases the grid is a rectangular grid which is deformed like a net that has stretched out of shape. Thus, each region has four sides, but the corners are not necessarily right angles. The method of interpolation is the simplest possible. First, we find the particular quadrilateral containing the point, as shown in figure 29. All that is necessary is to interpolate linearly within the quadrilateral. The interpolation is linear in the sense that it is linear along the boundary and between corresponding points along the boundary.

An illustration should clarify the manner of interpolation. Suppose it is desired to find the value at point $P$ in figure 30. It is assumed that values of the function are known at the corner points $A$, $B$, $C$, and $D$. The function values at these points will be designated $F_A$, $F_B$, $F_C$, and $F_D$. Suppose that the point $P$ lies on a line between points three-quarters of the way along $AB$ and $CD$, as shown. Also suppose that $P$ lies on a line between points two-thirds of the way along $BD$ and $AC$, as shown. Then, we can interpolate linearly along $AB$ and $CD$, followed by linear interpolation along the vertical line through $P$. If $F$ is the interpolated value of $P$, we obtain

$$F = \frac{1}{12} F_A + \frac{1}{4} F_B + \frac{1}{6} F_C + \frac{1}{2} F_D$$
The same result is obtained if we interpolate linearly along BD and AC, followed by linear interpolation along the horizontal line through P.

Figure 31 shows a quadrilateral containing a point $P_0$ where it is desired to interpolate. It is assumed that the values of the function to be interpolated are known at the four corners, and that the coordinates of the point $P_0$ are given. The function values are denoted by $z$, and the coordinates by $x$ and $y$. Subscripts are used to indicate the point. There are 14 values required to perform the interpolation: the coordinates of the four corners (eight values), the coordinates of the interpolation point (two values),
and the function values at the four corners. If it is assumed that these 14 values are known, an equation for linear interpolation can be derived.

Figure 32 shows the same quadrilateral as figure 31 but with the added lines $P_5P_6$ and $P_7P_8$. The line $P_5P_6$ passes through the point $P_0$ and is chosen so that $P_1P_5 : P_1P_3 = P_2P_6 : P_2P_4$. Similarly, $P_7P_8$ passes through $P_0$ and $P_1P_7 : P_1P_2 = P_3P_8 : P_3P_4$. Now, let

$$f_x = \frac{P_1P_7}{P_1P_2} \quad \text{(D1)}$$

$$f_y = \frac{P_1P_5}{P_1P_3} \quad \text{(D2)}$$

The coordinates of any point $P_i$ will be designated by $(x_i, y_i)$. The difference of any two $x$ or $y$ values will be designated by $x_{ij} = x_i - x_j$ or $y_{ij} = y_i - y_j$. Thus,

$$f_y = \frac{x_{51}}{x_{31}} = \frac{y_{51}}{y_{31}} = \frac{x_{62}}{x_{42}} = \frac{y_{62}}{y_{42}} \quad \text{(D3)}$$
The equation of line $P_5P_6$ is

$$\frac{y - y_5}{x - x_5} = \frac{y_{65}}{x_{65}} \tag{D4}$$

By using equation (D3), $y_5$, $y_6$, $x_5$, and $x_6$ can be expressed in terms of $f_y$ and the known values. For example,

$$y_5 = y_1 + y_{51} = y_1 - f_y y_{13}$$

In a similar manner we obtain

$$\begin{cases} y_5 = y_1 - f_y y_{13} \\ y_6 = y_2 + f_y y_{42} \\ x_5 = x_1 - f_y x_{13} \\ x_6 = x_2 + f_y x_{42} \end{cases} \tag{D5}$$

By substituting equations (D5) in (D4), we obtain

$$\frac{y - y_1 + f_y y_{13}}{x - x_1 + f_y x_{13}} = \frac{y_2 + f_y y_{42} - y_1 + f_y y_{13}}{x_2 + f_y x_{42} - x_1 + f_y x_{13}} \tag{D6}$$

This line passes through $P_0$, so when $x = x_0$, $y = y_0$. When this substitution is made and we multiply through by the denominators, we obtain a quadratic in $f_y$,

$$a f_y^2 + bf_y + c = 0 \tag{D7}$$

where

$$\begin{cases} a = y_{13} x_{42} - x_{13} y_{42} \\ b = x_{13} y_{02} - y_{13} x_{02} + y_{01} x_{42} - x_{01} y_{42} \\ c = y_{01} x_{21} - x_{01} y_{21} \end{cases} \tag{D8}$$
In a similar manner, we can obtain a quadratic in \( f_x \):

\[
af_x^2 + bf_x + c = 0
\]  \hspace{1cm} (D9)

where

\[
\begin{aligned}
a &= y_{12}x_{43} - x_{12}y_{43} \\
b &= x_{12}y_{03} - y_{12}x_{03} + y_{01}x_{43} - x_{01}y_{43} \\
c &= y_{01}x_{31} - x_{01}y_{31}
\end{aligned}
\]  \hspace{1cm} (D10)

If \( a \neq 0 \) in equation (D7) or (D9), there are two solutions for \( f_x \) or \( f_y \). However, there will be only one value between zero and one. When two sides are parallel, \( a \) will be zero and only one solution exists. Caution is needed when \( a \) is not zero but is very small. In this case there is one and only one solution between zero and one; but if the usual quadratic formula is used, the answer will be inaccurate. The solution, however, can be accurately calculated by using a binomial expansion.

If we let \( f \) represent either \( f_x \) or \( f_y \), the solution to either (D7) or (D9) can be written as

\[
f = \frac{-b}{2a} \left( 1 \pm \sqrt{1 - \frac{4ac}{b^2}} \right)
\]  \hspace{1cm} (D11)

When \( a \) is zero or small in magnitude, we want the root that is closest to zero. This is obtained by choosing the minus sign for the last term. Now we expand

\[
\left( 1 - \frac{4ac}{b^2} \right)^{1/2}
\]

by the binomial series, to obtain

\[
\sqrt{1 - \frac{4ac}{b^2}} = 1 - \frac{2ac}{b^2} - \frac{2a^2c^2}{b^4} - \frac{4a^3c^3}{b^6} - \frac{10a^4c^4}{b^8} - \ldots
\]  \hspace{1cm} (D12)
for $|4ac| < b^2$. Substituting equation (D12) in equation (D11) with the minus sign gives

$$f = -\frac{c}{b} \left( 1 + \frac{ac}{b^2} + \frac{2a^2c^2}{b^4} + \frac{5a^3c^3}{b^6} + \ldots \right)$$

(D13)

Equation (D13) is used when $ac/b^2$ is small. Otherwise, the usual quadratic formula is used. In the program (i.e., in subroutine LININT, and also in subroutine CONTIN), equation (D13) is used whenever $|4ac| \leq b^2/100$. Only three terms of the series are used; the term $5a^3c^3/b^6$ is dropped. This leads to a maximum relative error of less than $10^{-7}$. When $|4ac| > b^2/100$, the quadratic formula will lose no more than two or three decimal places in accuracy.

There is one further point that must be considered. Up to this point, it has been assumed that the interpolation point is within the overall grid area, and thus we only need to interpolate within a quadrilateral. However, there are cases where extrapolation is necessary. In this case, the nearest quadrilateral is identified, and extrapolation is used. The procedure is similar, but one of the f's must be either negative or greater than 1. The problem, then, is to determine which $f$ to use. Since the direction of the extrapolation is known, it is known whether $f$ is negative or greater than 1. For example, suppose it was necessary to extrapolate below the bottom of the grid area. Then $f_y$ must be negative. If only one of the two possible values is negative, the question is settled. If both are negative, the larger value (closest to zero) is used.

After both $f_x$ and $f_y$ are obtained, the linear interpolation can be performed to obtain $z_0$. Linear interpolation along $P_1P_2$ and $P_3P_4$ is followed by linear interpolation along $P_7P_8$. These interpolations are calculated by

$$z_7 = z_1 + f_x(z_2 - z_1)$$

$$z_8 = z_3 + f_x(z_4 - z_3)$$

$$z_0 = z_7 + f_y(z_8 - z_7)$$

Combining these equations, we get

$$z_0 = z_1(1 - f_x)(1 - f_y) + z_2f_x(1 - f_y) + z_3(1 - f_x)f_y + z_4f_xf_y$$

(D14)
APPENDIX E

SYMBOLS

\( B \)  
- tangential space between blades, corrected for loss of total pressure, rad

\( c_p \)  
- specific heat at constant pressure, J/(kg)(K)

\( F \)  
- vector normal to mid-channel stream surface and proportional to tangential pressure gradient, N/kg

\( H \)  
- absolute total enthalpy, J/kg

\( I \)  
- rothalpy, \( c_p T_1 - \omega \lambda \), meters\(^2\)/sec\(^2\)

\( m \)  
- meridional streamline distance, meters

\( p \)  
- pressure, N/meter\(^2\)

\( R \)  
- gas constant, J/(kg)(K)

\( r \)  
- radius from axis of rotation, meters

\( r_c \)  
- radius of curvature of meridional streamline, meters

\( s \)  
- distance along orthogonal mesh lines in throughflow direction (fig. 25), meters

\( T \)  
- temperature, K

\( t \)  
- distance along orthogonal mesh lines in direction across flow (fig. 25), meters

\( u \)  
- normalized stream function

\( V \)  
- absolute fluid velocity, meters/sec

\( W \)  
- fluid velocity relative to blade, meters/sec

\( W_{j+1} \)  
- \( W \) at next point, meters/sec

\( W^*_{j+1} \)  
- first estimate of \( W_{j+1} \), meters/sec

\( W^{**}_{j+1} \)  
- second estimate of \( W_{j+1} \), meters/sec

\( w \)  
- mass flow, kg/sec

\( z \)  
- axial coordinate, meters

\( \alpha \)  
- angle between meridional streamline and axis of rotation (fig. 2, part I), rad

\( \beta \)  
- angle between relative velocity vector and meridional plane (fig. 2, part I), rad

\( \gamma \)  
- specific heat ratio

\( \zeta \)  
- coefficient in stream-function equation, defined in eq. (A3), meters/sec\(^2\)

\( \theta \)  
- relative angular coordinate (fig. 2, part I), rad
\( \lambda \) prerotation, \( (rV_{1g})_1 \), meters\(^2\)/sec

\( \xi \) coefficient in stream-function equation, defined in eq. (A2), 1/meter

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

\( \varphi \) angle between s-coordinate line and axis of rotation (fig. 25), rad

\( \Omega \) overrelaxation factor

\( \omega \) rotational speed (fig. 2, part 1), rad/sec

Subscripts:

\( av \) average blade-to-blade value

\( b \) blade

\( bf \) blade flow

\( cr \) critical

\( fs \) free stream

\( hub \) hub

\( i \) inlet

\( l \) blade surface facing direction of positive rotation

\( le \) leading edge

\( m \) component in direction of meridional streamline

\( net \) net

\( o \) outlet

\( r \) component in radial direction

\( s \) component in s-direction

\( t \) component in t-direction

\( te \) trailing edge

\( tip \) tip

\( tr \) blade surface facing direction of negative rotation

\( z \) component in axial direction

\( \theta \) component in tangential direction

Superscripts:

\( ' \) absolute stagnation condition

\( '' \) relative stagnation condition

208
REFERENCES


☆ U.S. GOVERNMENT PRINTING OFFICE: 1974—729-159/198