FORTRAN PROGRAM FOR CALCULATING VELOCITIES IN A MAGNIFIED REGION ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE

by Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio
Page 45: Replace line 6 with
   IF (NOBL. EQ. 2) GO TO 95
   IF (MBII. LT. 1) MBII = 1000
   IF (MBOO. GT. MMM) MBOO = -1000
   GO TO 130

Page 46: The following lines should be inserted between statement 165 and the two calls on SPLINT:
   IF (MBII. GE. 1. AND. MBII. LE. MMM) MV(MBII) = MLE(3)
   IF (MBOO. GE. 1. AND. MBOO. LE. MMM) MV(MBOO) = CHORD(1) + MLE(1)
FORTRAN PROGRAM FOR CALCULATING VELOCITIES IN A MAGNIFIED REGION ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE

By Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio
ABSTRACT

A FORTRAN IV computer program was written to obtain a local detailed solution around a leading or trailing edge or in a slot region for compressible, subsonic, nonviscous flow on a blade-to-blade surface between turbomachine blades. This program allows a coarse-mesh solution for an entire blade-to-blade region to be magnified in a small rectangular region. The results include detailed surface velocities, velocity magnitude and direction, and stream-function values throughout the magnified region. The method is based on the stream function and uses the iterative solution of nonlinear finite-difference equations.
## 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>1</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>2</td>
</tr>
<tr>
<td>MATHEMATICAL ANALYSIS</td>
<td>3</td>
</tr>
<tr>
<td>NUMERICAL EXAMPLES</td>
<td>4</td>
</tr>
<tr>
<td>Leading Edge of Rear Blade of Tandem Blade Turbine Rotor</td>
<td>4</td>
</tr>
<tr>
<td>Trailing Edge of Mixed-Flow Impeller</td>
<td>5</td>
</tr>
<tr>
<td>DESCRIPTION OF INPUT AND OUTPUT</td>
<td>13</td>
</tr>
<tr>
<td>Instructions for Preparing Input</td>
<td>13</td>
</tr>
<tr>
<td>Modification of 2DCP Input</td>
<td>13</td>
</tr>
<tr>
<td>Choosing Magnified Region</td>
<td>16</td>
</tr>
<tr>
<td>Input</td>
<td>17</td>
</tr>
<tr>
<td>Example of Preparing Input</td>
<td>18</td>
</tr>
<tr>
<td>Output</td>
<td>25</td>
</tr>
<tr>
<td>Error Conditions</td>
<td>26</td>
</tr>
<tr>
<td>PROGRAM PROCEDURE</td>
<td>28</td>
</tr>
<tr>
<td>Conventions Used in Program</td>
<td>30</td>
</tr>
<tr>
<td>Labeled COMMON Blocks</td>
<td>30</td>
</tr>
<tr>
<td>Subroutine INPUT</td>
<td>31</td>
</tr>
<tr>
<td>Subroutine PRECAL</td>
<td>31</td>
</tr>
<tr>
<td>Subroutine WRITU</td>
<td>31</td>
</tr>
<tr>
<td>Main Dictionary</td>
<td>32</td>
</tr>
<tr>
<td>Program Listing for Subroutines Using Main Dictionary</td>
<td>41</td>
</tr>
<tr>
<td>Subroutine BDVINT</td>
<td>73</td>
</tr>
<tr>
<td>Program Listing of Remaining Subroutines</td>
<td>75</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>80</td>
</tr>
</tbody>
</table>
FORTRAN PROGRAM FOR CALCULATING VELOCITIES IN A MAGNIFIED REGION
ON A BLADE-TO-BLADE STREAM SURFACE OF A TURBOMACHINE
by Theodore Katsanis and William D. McNally
Lewis Research Center

SUMMARY

A FORTRAN IV computer program was written to obtain a local detailed solution around a leading or trailing edge or in a slot region for compressible, subsonic, non-viscous flow on a blade-to-blade surface between turbomachine blades. This program allows a coarse-mesh solution for an entire blade-to-blade region to be magnified by a chosen magnification factor in a small rectangular region.

The program input requires information obtained from a less detailed solution from one of three other FORTRAN programs. These programs have been presented in NASA Technical Notes. The output includes detailed surface velocities, velocity magnitude and direction, and stream-function values throughout the magnified region.

The method is based on the stream function with the solution of the simultaneous, nonlinear, finite-difference equations being obtained by two major levels of iteration. The inner iteration consists of the solution of simultaneous linear equations by successive overrelaxation, using an estimated optimum overrelaxation factor. The outer iteration then changes the coefficient of the simultaneous equations to compensate for compressibility.

This report includes the FORTRAN IV computer program with an explanation of the equations involved, the method of solution, and the calculation of velocities. Numerical examples have been included to illustrate the use of the program and to show the results which are obtained.

INTRODUCTION

In the design of blade rows for turbines and compressors, it is desirable to obtain the velocity distribution through the passage and particularly over the blade surfaces. The authors have published computer programs (refs. 1 to 3) for obtaining this type of
solution for single and tandem blade rows.

With these programs, however, it is not always possible to obtain sufficient detail on some critical parts of the blade surfaces. These programs give an approximate solution for velocities only at the mesh points of a finite-difference grid. Due to storage requirements on the computer, grid spacing may be too large to give the desired detail around small leading- or trailing-edge radii or within slot regions. And it is in these regions where geometry, and thus velocities, change most rapidly.

For these reasons a computer program has been written to obtain a solution on a fine mesh in a small part of the blade-to-blade region. The method used is similar to that used by Kramer (ref. 4). A small rectangular region of the solution obtained by either 2DCP (ref. 1), TURBLE (ref. 2), or TANDEM (ref. 3) can be magnified by a chosen factor using MAGNFY, the program described herein. The input and output are similar to 2DCP, TURBLE, and TANDEM, with additional input required. The additional input is obtained from the output of 2DCP, TURBLE, or TANDEM.

This report includes the FORTRAN IV computer program that was developed with explanation of the input required. An axial-flow turbine rotor slot and the tip of a mixed-flow impeller have been analyzed to illustrate the use of the program.

This report is organized so that the engineer desiring to use this program needs to read only the sections MATHEMATICAL ANALYSIS, NUMERICAL EXAMPLES, and DESCRIPTION OF INPUT AND OUTPUT. The necessary information of interest to a programmer is contained in the sections DESCRIPTION OF INPUT AND OUTPUT and PROGRAM PROCEDURE.

A MAGNFY source deck on tape is available from COSMIC (Computer Software Management and Information Center), Computer Center, University of Georgia, Athens, Georgia 30601. The program number is COSMIC number LEW-10789.

**SYMBOLS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>coefficient matrix, (eq. (A7), ref. 1)</td>
</tr>
<tr>
<td>b</td>
<td>stream-channel thickness normal to meridional streamline, meters</td>
</tr>
<tr>
<td>k</td>
<td>constant vector, $\begin{pmatrix} k_1 \ \vdots \ k_n \end{pmatrix}$, (eq. (A7), ref. 1)</td>
</tr>
<tr>
<td>m</td>
<td>meridional streamline distance, meters</td>
</tr>
<tr>
<td>R</td>
<td>gas constant, joule/(kg)($^\circ$K)</td>
</tr>
</tbody>
</table>
MATHEMATICAL ANALYSIS

It is desired to determine the flow distribution over the leading or trailing edge of a turbomachine blade or through the slot of a tandem or slotted blade. The stream function is used for the analysis. The basic assumptions and equations are given in references 1 and 3. The only difference in this analysis from that of references 1 and 3 is in the boundary conditions. For the MAGNY program, the value of the stream function must be given for the entire boundary of the region considered. These values are determined on a coarse mesh by 2DCP, TURBLE, or TANDEM. MAGNY then interpolates
these values to obtain boundary values of stream functions on a finer mesh. These boundary conditions determine a solution to the stream function (eq. (1), ref. 1). The numerical solution is determined by using finite-difference equations, as described in appendix A of references 1 and 3.

NUMERICAL EXAMPLES

Two numerical examples are given to illustrate the use of the program and to show the type of results which can be obtained. The first example is the slot region of a tandem axial-flow turbine rotor blade, and the other is the trailing edge of a mixed-flow impeller.

Leading Edge of Rear Blade of Tandem Blade Turbine Rotor

Flow about the leading edge of the rear blade of a tandem axial-flow turbine rotor blade (ref. 5) has been analyzed to illustrate the use of MAGNFY. The entire blade was first analyzed by using TANDEM with the mesh size shown in figure 1. Due to computer storage limitations, this was as fine a mesh as could be obtained with TANDEM. However, more detail was desired for velocities between adjacent mesh points on the leading edge of the tandem blade. Therefore, MAGNFY was used in order to reduce the mesh spacing in the region around the leading edge by a factor of 8, as shown in figure 2.

The input for this case is given in table I. This includes most of the input necessary
for TANDEM, in addition to the stream-function values on the vertical and horizontal boundaries of the magnified region. These stream-function values and their coordinates were obtained as output from the TANDEM program.

Blade-surface velocities from both TANDEM and MAGNFY are plotted in figure 3. The original velocities obtained by using the coarse mesh are denoted by circles in figure 3. The MAGNFY output is plotted as a solid line. As shown in figure 3, the velocity peak on the suction surface is much higher than indicated by the coarse-mesh solution. This illustrates the need for a finer grid in some parts of the solution region since the velocities denoted by circles do not define the velocity adequately.

The execution time for this example was 10 minutes on the direct-coupled IBM2-7094-7044 computer.

![Graph comparing velocities from coarse-mesh and fine-mesh solutions.](image)

**Figure 3.** Comparison of velocities from coarse-mesh and fine-mesh solutions.

**Trailing Edge of Mixed-Flow Impeller**

Flow about the trailing edge of the main blade of a mixed-flow impeller with splitter blades (ref. 6) has also been analyzed using MAGNFY. The entire impeller was originally analyzed with TANDEM (ref. 3). The impeller profile (in the meridional plane) is shown in figure 4(a). Figure 4(b) shows the blade-to-blade region with the coarse mesh used in the TANDEM run. The region to be magnified about the trailing edge of the main blade is indicated by heavy lines in figure 4(b). MAGNFY was used to reduce the mesh.
<table>
<thead>
<tr>
<th>TABLE I. - MAGNIFY INPUT FOR LEADING EDGE OF REAR BLADE OF TANDEM BLADE TURBINE ROTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIFTED TANDEM AXIAL TURBINE ROTOR - SMALL SLOT AREA</td>
</tr>
<tr>
<td>NOPN BLADE SURFACE 1 -- UPPER SURFACE - FRONT BLADE</td>
</tr>
<tr>
<td>R11 33 33 0.762000E-03 0.3810000E-03 50.000000 -29.400000 7.0000000</td>
</tr>
<tr>
<td>MSP1 ARRAY -0 THSP1 ARRAY 0.9250000E-02 0.21180000E-01 0.29880000E-01 0.30200000E-01</td>
</tr>
<tr>
<td>-0 -0 0.2570000E-02 0.3810000E-03 -0.7650000E-02 -0.2750000E-01 0.2130000E-01 0.2540000E-01 0.2640000E-01</td>
</tr>
<tr>
<td>BLADE SURFACE 2 -- LOWER SURFACE - FRONT BLADE</td>
</tr>
<tr>
<td>R12 33 33 0.762000E-03 0.3810000E-03 25.000000 -6.9000000 5.0000000</td>
</tr>
<tr>
<td>MSP2 ARRAY -0 THSP2 ARRAY 0.7140000E-02 0.20390000E-01 0.2540000E-01 0.2640000E-01</td>
</tr>
<tr>
<td>-0 -0 -0.7650000E-02 -0.2750000E-01 -0.2130000E-01 -0.2540000E-01 -0.2640000E-01</td>
</tr>
<tr>
<td>BLADE SURFACE 3 -- UPPER SURFACE - REAR BLADE</td>
</tr>
<tr>
<td>R13 33 303 0.762000E-03 0.3810000E-03 13.000000 -48.800000 5.0000000</td>
</tr>
<tr>
<td>MSP3 ARRAY -0 THSP3 ARRAY 0.4080000E-02 -0.2400000E-03 -0.2651000E-01</td>
</tr>
<tr>
<td>-0 -0 0.5160000E-02 0.12780000E-01 0.22940000E-01 0.2040000E-01 0.2933000E-01</td>
</tr>
<tr>
<td>BLADE SURFACE 4 -- LOWER SURFACE - REAR BLADE</td>
</tr>
<tr>
<td>R14 33 33 0.762000E-03 0.3810000E-03 -4.2000000 -42.500000 5.0000000</td>
</tr>
<tr>
<td>MSP4 ARRAY -0 THSP4 ARRAY -0.4390000E-02 -0.1388000E-01 -0.2933000E-01</td>
</tr>
<tr>
<td>-0 -0 0.5160000E-02 0.12780000E-01 0.22940000E-01 0.2040000E-01 0.2933000E-01</td>
</tr>
<tr>
<td>MR ARRAY -1.0000000 1.0000000</td>
</tr>
<tr>
<td>RMS ARRAY 0.3236 0.3236</td>
</tr>
<tr>
<td>DESP ARRAY 0.1143 0.1143</td>
</tr>
<tr>
<td>BLDAT AANDK EEKOR STRFN INTVL SURVL</td>
</tr>
<tr>
<td>1 1 2 2 2 3</td>
</tr>
<tr>
<td>KBDRY NSP</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
(a) Hub-shroud profile, showing meridional section of stream tube.

Figure 4. - Mixed-flow pump impeller,
(b) Blade-to-blade surface, showing coarse grid used in TANDEM program.

Figure 4. - Concluded.
TABLE II. MAGNIFY INPUT FOR TRAILING EDGE OF MIXED-FLOW IMPELLER

<table>
<thead>
<tr>
<th>NOBL</th>
<th>GAM</th>
<th>AR</th>
<th>TIP</th>
<th>RHQIP</th>
<th>WTFL</th>
<th>WTFLSP</th>
<th>OMEGA</th>
<th>ORF</th>
<th>QRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.500000</td>
<td>1000.0000</td>
<td>1000000.0</td>
<td>1.000000</td>
<td>0.3042000E-02</td>
<td>0.1351600E-02</td>
<td>796.00000</td>
<td>1.0440000</td>
<td>84.880000</td>
</tr>
<tr>
<td></td>
<td>BETA0</td>
<td>BETA1</td>
<td>CHORDF</td>
<td>STGRF</td>
<td>CHOROR</td>
<td>STGRR</td>
<td>THLRR</td>
<td>MLER</td>
<td>TMLER</td>
</tr>
<tr>
<td></td>
<td>-43.000000</td>
<td>-84.880000</td>
<td>0.1055500</td>
<td>-2.6290000</td>
<td>0.5664000E-01</td>
<td>-0.6649000</td>
<td>0.4891000E-01</td>
<td>2.3434000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MB1</th>
<th>MB0</th>
<th>MB12</th>
<th>MB02</th>
<th>MM</th>
<th>NBBI</th>
<th>NBL</th>
<th>NRSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>47</td>
<td>28</td>
<td>47</td>
<td>57</td>
<td>28</td>
<td>8</td>
<td>18</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 1 -- UPPER SURFACE - FRONT BLADE**

<table>
<thead>
<tr>
<th>R12</th>
<th>RO1</th>
<th>BETA1</th>
<th>BET01</th>
<th>SPLN01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1328000E-02</td>
<td>0.1753000E-02</td>
<td>-60.500000</td>
<td>-51.500000</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 2 -- LOWER SURFACE - FRONT BLADE**

<table>
<thead>
<tr>
<th>R12</th>
<th>RO2</th>
<th>BETA2</th>
<th>BET02</th>
<th>SPLN02</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1328000E-02</td>
<td>0.1753000E-02</td>
<td>-60.500000</td>
<td>-51.500000</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 3 -- UPPER SURFACE - REAR BLADE**

<table>
<thead>
<tr>
<th>R13</th>
<th>RO3</th>
<th>BETA3</th>
<th>BET03</th>
<th>SPLN03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1328000E-02</td>
<td>0.1753000E-02</td>
<td>-60.500000</td>
<td>-51.500000</td>
</tr>
</tbody>
</table>

**BLADE SURFACE 4 -- LOWER SURFACE - REAR BLADE**

<table>
<thead>
<tr>
<th>R14</th>
<th>RO4</th>
<th>BETA4</th>
<th>BET04</th>
<th>SPLN04</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.1328000E-02</td>
<td>0.1753000E-02</td>
<td>-60.500000</td>
<td>-51.500000</td>
</tr>
</tbody>
</table>

**MR ARRAY**

| -0.3124000E-01 | -0.1514000E-01 | 0.2500000E-03 | 0.1065000E-01 | 0.1853000E-01 | 0.2651000E-01 | 0.3460000E-01 | 0.4281000E-01 |
| 0.5115000E-01 | 0.5964000E-01 | 0.6828000E-01 | 0.7709000E-01 | 0.8607000E-01 | 0.8530000E-01 | 0.8530000E-01 | 0.8530000E-01 |
| 0.1272000 | 0.1407300 | 0.1753000E-02 | 0.8740000E-01 | 0.4281000E-01 | 0.8530000E-01 | 0.8530000E-01 | 0.8530000E-01 |
| 0.7596000E-01 | 0.7662000E-01 | 0.8740000E-01 | 0.8740000E-01 | 0.8740000E-01 | 0.8740000E-01 | 0.8740000E-01 | 0.8740000E-01 |
| 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 | 0.9447000E-01 |
| 0.1360200 | 0.1448700 | 0.1448700 | 0.1448700 | 0.1448700 | 0.1448700 | 0.1448700 | 0.1448700 |

<p>| 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 | 0.1053000E-01 |
| 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 | 0.3235000E-02 |
| 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 | 0.6250000E-02 |</p>
<table>
<thead>
<tr>
<th>BLDAT</th>
<th>AANDK</th>
<th>ERSOR</th>
<th>STRFN</th>
<th>INTVL</th>
<th>SURVL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MBDFY</th>
<th>MBDFYL</th>
<th>ITF</th>
<th>ITL</th>
<th>MAGFAC</th>
<th>LAMBD.A</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>50</td>
<td>-97</td>
<td>-90</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

-0.9835000E-02

<table>
<thead>
<tr>
<th>KBDRY NSP</th>
<th>1</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVIN ARRAY</td>
<td>0.9660700E-01</td>
<td>0.9958800E-01</td>
</tr>
<tr>
<td>UBVIN ARRAY</td>
<td>-0.2089400</td>
<td>-0.1757100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KBDRY NSP</th>
<th>2</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVIN ARRAY</td>
<td>0.9660700E-01</td>
<td>0.9958800E-01</td>
</tr>
<tr>
<td>UBVIN ARRAY</td>
<td>0.1517000E-01</td>
<td>0.4847000E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KBDRY NSP</th>
<th>3</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVIN ARRAY</td>
<td>-2.7208400</td>
<td>-2.6927900</td>
</tr>
<tr>
<td>UBVIN ARRAY</td>
<td>-0.2089400</td>
<td>-0.1676100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KBDRY NSP</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVIN ARRAY</td>
<td>-2.5380100</td>
<td>-2.5244900</td>
</tr>
<tr>
<td>UBVIN ARRAY</td>
<td>0.1517000E-01</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KBDRY NSP</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBVIN ARRAY</td>
<td>-0.2077000E-01</td>
<td>0.1055500</td>
</tr>
</tbody>
</table>
size in this region by a factor of 5. The reduced mesh is shown in figure 5.

The input for this example is given in table II. It includes the original TANDEM input plus stream-function boundary values about the magnified region. These were obtained from the output of TANDEM. The process by which this input was obtained is explained later in the section Example of Preparing Input.

The blade-surface velocities from TANDEM and MAGNFY are plotted in figure 6. The velocities obtain by TANDEM (coarse mesh) are not accurate near the trailing-edge radius, and the magnitude of the peak is not shown accurately.

The execution time for this example was 2 minutes on the direct-coupled IBM2-7094-7044 computer.
DESCRIPTION OF INPUT AND OUTPUT

The computer program requires as input the same input as was used in either 2DCP (ref. 1), TURBLE (ref. 2), or TANDEM (ref. 3) plus the stream-function values along the boundary of the region to be magnified. These stream-function values are generally obtained from the output of either 2DCP, TURBLE, or TANDEM (refs. 1 to 3).

Output obtained from MAGNFY includes velocity magnitude and direction at all interior mesh points in the region, blade-surface velocities, and stream-function values throughout the region.

Instructions for Preparing Input

The first step in obtaining input for MAGNFY is to obtain the usual coarse-mesh solution from either 2DCP (ref. 1), TURBLE (ref. 2), or TANDEM (ref. 3). If TURBLE or TANDEM are used, their input forms part of the input for MAGNFY. If 2DCP is used, modifications must be made to its input to make it appropriate for MAGNFY. These modifications make the 2DCP input agree with input for TURBLE, and are explained in the following section.

The remainder of the input for MAGNFY (beyond the input for 2DCP, TURBLE, or TANDEM) consists of coordinates and stream-function values obtained from the output of one of these three programs. Figure 7 shows all the input for MAGNFY beyond that required for 2DCP, TURBLE, or TANDEM.

The boundary value input for MAGNFY could be determined from some method other than either 2DCP, TURBLE, or TANDEM. In this case the input which would ordinarily have been used with these programs must be determined as explained in references 1 or 2 (single blade) or reference 3 (tandem or slotted blade).

Modification of 2DCP Input

If the user desires to magnify a solution obtained with 2DCP, the 2DCP input must be rearranged as if it were to be run on TURBLE before it can be used with MAGNFY. Some of the 2DCP input variables have the same names as MAGNFY variables, but some
Figure 7. - Input form. Card column numbers appear at top.
do not. Table III lists the 2DCP and MAGNY variables which have the same meaning but different names. Figure 8 and table III show the user how to rearrange his 2DCP input to make it compatible with MAGNY. The first card of 2DCP input (the GAM card) must be modified for MAGNY by shifting OMEGA and W 10 spaces to the right (see fig. 8). The second card should have the inlet and outlet flow angles (BETAI and BETAO) placed before CHORD and STGR. Also BETAI and BETAO have been redefined to be the flow angles at the leading and trailing edges, instead of at upstream and downstream boundaries. The third card contains information obtained from the fourth card for 2DCP. Once again, the position of variables on the card and the relation between 2DCP and MAGNY variables can be seen from figure 8 and table III. The information on the third 2DCP card must be placed on two cards for MAGNY. These cards are placed directly above the two sets of m- and θ-coordinates (MSP1,2 and THSP1,2) for the two blade surfaces. Each of these cards for MAGNY contains inlet and outlet radii, tangency angles, and number of spline points for one of the two blade surfaces. Finally, the cards containing m- and θ-coordinates are unchanged between 2DCP and MAGNY.
TABLE III. - 2DCP AND MAGNY VARIABLES WITH
SAME MEANING BUT DIFFERENT NAMES

<table>
<thead>
<tr>
<th>2DCP variable</th>
<th>Corresponding MAGNY variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>ORF</td>
</tr>
<tr>
<td>RI</td>
<td>RI1 and RI2</td>
</tr>
<tr>
<td>RO</td>
<td>RO1 and RO2</td>
</tr>
<tr>
<td>ALUI</td>
<td>BETI1</td>
</tr>
<tr>
<td>ALLI</td>
<td>BETI2</td>
</tr>
<tr>
<td>ALUO</td>
<td>BETO1</td>
</tr>
<tr>
<td>ALLO</td>
<td>BETO2</td>
</tr>
<tr>
<td>MXBI</td>
<td>MBI</td>
</tr>
<tr>
<td>MXBO</td>
<td>MBO</td>
</tr>
<tr>
<td>MX</td>
<td>MM</td>
</tr>
<tr>
<td>NUSP</td>
<td>SPLNO1 (real)</td>
</tr>
<tr>
<td>NLSP</td>
<td>SPLNO2 (real)</td>
</tr>
<tr>
<td>MU</td>
<td>MSP1</td>
</tr>
<tr>
<td>XSPU</td>
<td>THSP1</td>
</tr>
<tr>
<td>ML</td>
<td>MSP2</td>
</tr>
<tr>
<td>XSPL</td>
<td>THSP2</td>
</tr>
</tbody>
</table>

Choosing Magnified Region

The region where magnification is desired is usually located around a leading or trailing edge or about a tandem blade slot. Therefore, the region generally includes portions of both lower and upper blade surfaces. This is indicated by the heavily outlined region of figure 9 and by the similar region for the mixed-flow impeller example (fig. 4(b)). However, input must be given to MAGNY as though the region was entirely located about the lower blade. (See the dashed portion of the region in fig. 9.) This condition, of course, results in a magnified region which is partially outside of the original 2DCP, TURBLE, or TANDEM region. In the case of the leading edge of the rear blade of a tandem blade, the magnified region may lie completely outside of the original TANDEM region. The region may contain at most one leading and one trailing edge.

The fact that the magnified region is restricted to the lower blade implies that, once it is drawn about the lower blade, no part of it may include any of the upper blade.
Input

Only the additional input beyond that required for 2DCP, TURBLE, or TANDEM will be described in detail herein. All integers are in a five-column field and must be right adjusted. All numbers with a ten-column field are real numbers and must have a decimal point punched on the data card.

The first input data card (fig. 7) is a label card containing any desired identification label. The next card has either NOBL=1 (single blade) or NOBL=2 (tandem or slotted blade) in column 5. This is followed by the input data from either TURBLE or TANDEM, or modified input from 2DCP. This input consists of all the input cards from the first one starting with GAM up to the last geometry input card containing values of BESP. This input remains unchanged, except for the overrelaxation factor (ORF), which should be recalculated for the MAGNFY program; that is, it should be set equal to zero again for the initial MAGNFY run on a set of data.

The next input card has variables (BLDAT to SURVL) used to indicate what output is desired. These variables are used in the same way as in TURBLE or TANDEM, except for the ommission of SLCRD, which is not required in MAGNFY.
The remaining input variables (see fig. 7) are as follows:

- **MBDYF**: index IM of vertical mesh line which is to be left boundary of magnified region (see fig. 9)
- **MBDYL**: index IM of vertical mesh line which is to be right boundary of magnified region
- **ITF**: index IT of horizontal mesh line which is to be lower boundary of magnified region (See previous section for explanation of how to choose magnified region.)
- **ITL**: index IT of horizontal mesh line which is to be upper boundary of magnified region
- **MAGFAC**: magnification factor (If MAGFAC = n, one mesh square of original coarse mesh will contain $n^2$ squares of smaller mesh.)
- **LAMBDA**: value of prerotation $\lambda$ at inlet (LAMBDA is given as part of the output for 2DCP, TURBLE, or TANDEM.)
- **KBDRY**: indicates which boundary is referred to on input cards which follow it:
  - KBDRY=1, lower boundary
  - KBDRY=2, upper boundary
  - KBDRY=3, left boundary
  - KBDRY=4, right boundary
- **NSP**: number of stream-function values given for a particular boundary on BVIN and UBVIN cards which follow it
- **BVIN**: boundary coordinates (m or $\theta$) for segment of boundary indicated by KBDRY (These coordinates should correspond to original coarse-mesh lines, except for possibly the first and last points, which could fall on a blade surface.)
- **UBVIN**: stream-function values corresponding to BVIN

The variables from KBDRY to UBVIN are given for each segment of the boundary. After all boundary values are given for each segment of the four boundaries, one blank card (or a card with zeros for KBDRY and NSP) is added to signal the end of the input data for a particular case.

**Example of Preparing Input**

The second numerical example of this report (p. 5) dealt with solving for detailed
Figure 10. - Mixed-flow impeller, showing information for input example.
velocities about the trailing edge of the main blade of a mixed-flow impeller with splitter vanes (ref. 6). This section illustrates, in detail, how the additional MAGNFY input (beyond the normal TANDEM input) for that example was obtained. This input is given in table II.

MAGNFY is intended to be used by those who have run 2DCP, TURBLE, or TANDEM, and who desire a more detailed solution about some critical region on the blade. For the impeller example, that region is the trailing edge of the main blade as shown in figure 10. Notice that most of this region lies about blade surface 1, but that a portion of it is located at the end of blade surface 2. In most cases, the region to be analyzed will be divided in this way. However, input must be given to MAGNFY as though the region is entirely located about the lower blade. The way this is done is illustrated in the following paragraph.

The user should draw a magnified picture of the region for which a detailed solution is desired (fig. 11). This rectangular picture should extend three or four mesh lines (coarse mesh) in all directions from the point at which most detail is desired. The coarse mesh should be numbered with IM and IT grid line values. The ITV array of TURBLE or TANDEM can be used in the drawing of this sketch. The boundaries parallel to the IM axis are defined as boundaries 1 and 2 of this sketch, and those parallel to the IT axis as 3 and 4, as indicated in figure 11.

At this point, some of the input to MAGNFY can be obtained. The first and last values on the IM and IT axes are called MBDYF, MBDYL, ITF, and ITL. For the impeller example, these values are 44, 50, -97, and -90, respectively.
MAGFAC can also be chosen at this time. A number between 5 and 8 is typical, and in this case 5 was used. MAGFAC must be chosen so that the resulting grid has less than 2000 mesh points in it.

LAMBDA is the next required input. It is obtained directly from the output of either 2DCP, TURBLE, or TANDEM.

The remainder of the MAGNFY input consists of geometrical (BVIN) and stream-function (UBVIN) boundary values for the coarse-mesh boundary points on the four boundaries of the magnified region (fig. 11). The boundaries 1 to 4 should be entered in order, giving values from left to right on boundaries 1 and 2 and from bottom to top on boundaries 3 and 4.

The blade surfaces always intersect some boundaries of the region. When this occurs, the resulting sections of boundaries should be entered separately. In this example, boundary 3 is divided into two parts. The first has six points (IT = -97 to IT = -93 plus the blade-surface point (point a), fig. 11). The second has two points (point b and IT = -90).

For each section of a boundary, four items of input are needed: KBDRY, NSP, the BVIN array, and the UBVIN array. Data are given, and NSP is the number of points given on a section of that boundary. For the impeller example, NSP = 7 on boundaries 1 and 2; NSP = 6 and NSP = 2 on boundary 3; and NSP = 8 on boundary 4.

For boundaries 1 and 2, BVIN is obtained from the Stream Sheet Coordinates and Thickness Table for both TURBLE and TANDEM. A portion of the table for this example is reproduced in table IV. The meridional coordinates for BVIN from IM = 44 to IM = 50 are circled.

<table>
<thead>
<tr>
<th>IM</th>
<th>m</th>
<th>R</th>
<th>SAL</th>
<th>B</th>
<th>DB/DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2445E-01</td>
<td>0.76328E-01</td>
<td>0.38990E-01</td>
<td>0.10422E-01</td>
<td>-0.25197E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.2173E-01</td>
<td>0.76152E-01</td>
<td>0.52336E-01</td>
<td>0.10342E-01</td>
<td>-0.33504E-01</td>
</tr>
<tr>
<td>3</td>
<td>0.1902E-01</td>
<td>0.76314E-01</td>
<td>0.67178E-01</td>
<td>0.10239E-01</td>
<td>-0.42743E-01</td>
</tr>
<tr>
<td>4</td>
<td>0.1630E-01</td>
<td>0.76519E-01</td>
<td>0.83516E-01</td>
<td>0.10109E-01</td>
<td>-0.52913E-01</td>
</tr>
<tr>
<td>5</td>
<td>0.1358E-01</td>
<td>0.76769E-01</td>
<td>0.10100E-01</td>
<td>0.99507E-02</td>
<td>-0.63772E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IM</th>
<th>m</th>
<th>R</th>
<th>SAL</th>
<th>B</th>
<th>DB/DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>0.8170E-01</td>
<td>0.10914</td>
<td>0.52474</td>
<td>0.17711E-02</td>
<td>-0.34060E-01</td>
</tr>
<tr>
<td>40</td>
<td>0.8468E-01</td>
<td>0.11071</td>
<td>0.53430</td>
<td>0.16727E-02</td>
<td>-0.31962E-01</td>
</tr>
<tr>
<td>41</td>
<td>0.8766E-01</td>
<td>0.11233</td>
<td>0.54631</td>
<td>0.15804E-02</td>
<td>-0.29980E-01</td>
</tr>
<tr>
<td>42</td>
<td>0.9064E-01</td>
<td>0.11397</td>
<td>0.55690</td>
<td>0.14938E-02</td>
<td>-0.28176E-01</td>
</tr>
<tr>
<td>43</td>
<td>0.9362E-01</td>
<td>0.11564</td>
<td>0.56553</td>
<td>0.14122E-02</td>
<td>-0.26560E-01</td>
</tr>
<tr>
<td>44</td>
<td>0.9660E-01</td>
<td>0.11734</td>
<td>0.57246</td>
<td>0.13352E-02</td>
<td>-0.25103E-01</td>
</tr>
<tr>
<td>45</td>
<td>0.9958E-01</td>
<td>0.11906</td>
<td>0.57954</td>
<td>0.12626E-02</td>
<td>-0.23590E-01</td>
</tr>
<tr>
<td>46</td>
<td>0.10259</td>
<td>0.12080</td>
<td>0.58712</td>
<td>0.11974E-02</td>
<td>-0.21979E-01</td>
</tr>
<tr>
<td>47</td>
<td>0.10559</td>
<td>0.12256</td>
<td>0.59514</td>
<td>0.11317E-02</td>
<td>-0.20275E-01</td>
</tr>
<tr>
<td>48</td>
<td>0.10853</td>
<td>0.12434</td>
<td>0.60279</td>
<td>0.10739E-02</td>
<td>-0.18520E-01</td>
</tr>
<tr>
<td>49</td>
<td>0.11151</td>
<td>0.12615</td>
<td>0.60980</td>
<td>0.10213E-02</td>
<td>-0.16728E-01</td>
</tr>
<tr>
<td>50</td>
<td>0.11449</td>
<td>0.12798</td>
<td>0.61617</td>
<td>0.97415E-03</td>
<td>-0.14901E-01</td>
</tr>
<tr>
<td>51</td>
<td>0.11747</td>
<td>0.12983</td>
<td>0.62238</td>
<td>0.93233E-03</td>
<td>-0.13193E-01</td>
</tr>
<tr>
<td>52</td>
<td>0.12046</td>
<td>0.13169</td>
<td>0.62878</td>
<td>0.89527E-03</td>
<td>-0.11706E-01</td>
</tr>
<tr>
<td>53</td>
<td>0.12344</td>
<td>0.13357</td>
<td>0.63535</td>
<td>0.86232E-03</td>
<td>-0.10440E-01</td>
</tr>
<tr>
<td>54</td>
<td>0.12642</td>
<td>0.13548</td>
<td>0.64210</td>
<td>0.83281E-03</td>
<td>-0.09395E-02</td>
</tr>
<tr>
<td>55</td>
<td>0.12940</td>
<td>0.13740</td>
<td>0.64879</td>
<td>0.80611E-03</td>
<td>-0.08539E-02</td>
</tr>
</tbody>
</table>

TABLE IV. - STREAM SHEET COORDINATES AND THICKNESS TABLE
TABLE V. - THETA COORDINATES OF HORIZONTAL MESH LINES

<table>
<thead>
<tr>
<th>IT</th>
<th>THETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>-107</td>
<td>-3.00134</td>
</tr>
<tr>
<td>-106</td>
<td>-2.97329</td>
</tr>
<tr>
<td>-105</td>
<td>-2.94524</td>
</tr>
<tr>
<td>-104</td>
<td>-2.91719</td>
</tr>
<tr>
<td>-103</td>
<td>-2.88914</td>
</tr>
<tr>
<td>-102</td>
<td>-2.86109</td>
</tr>
<tr>
<td>-101</td>
<td>-2.83304</td>
</tr>
<tr>
<td>-100</td>
<td>-2.80499</td>
</tr>
<tr>
<td>-99</td>
<td>-2.77694</td>
</tr>
<tr>
<td>-98</td>
<td>-2.74889</td>
</tr>
<tr>
<td>-97</td>
<td>-2.72084</td>
</tr>
<tr>
<td>-96</td>
<td>-2.69279</td>
</tr>
<tr>
<td>-95</td>
<td>-2.66474</td>
</tr>
<tr>
<td>-94</td>
<td>-2.63669</td>
</tr>
<tr>
<td>-93</td>
<td>-2.60864</td>
</tr>
<tr>
<td>-92</td>
<td>-2.58059</td>
</tr>
<tr>
<td>-91</td>
<td>-2.55254</td>
</tr>
<tr>
<td>-90</td>
<td>-2.52449</td>
</tr>
<tr>
<td>-89</td>
<td>-2.49644</td>
</tr>
<tr>
<td>-88</td>
<td>-2.46839</td>
</tr>
<tr>
<td>-87</td>
<td>-2.44034</td>
</tr>
<tr>
<td>-86</td>
<td>-2.41229</td>
</tr>
<tr>
<td>-85</td>
<td>-2.38424</td>
</tr>
<tr>
<td>-84</td>
<td>-2.35619</td>
</tr>
</tbody>
</table>

If a blade surface passes through boundary 1 or 2 (it does not in this example), the BVIN for the point of intersection of the blade and boundary is obtained from the MH array (m-coordinates of intersections of horizontal mesh lines with blade given as output from 2DCP, TANDEM, or TURBLE) for the blade surface involved.

For boundaries 3 and 4, BVIN is obtained from the table of Theta Coordinates of Horizontal Mesh Lines for both TURBLE and TANDEM. A portion of this table for this example is reproduced in table V. The tangential coordinates for BVIN from IT = -97 to IT = -90 are circled.

If a blade surface passes through boundary 3 or 4, the BVIN for the point of intersection of the blade and boundary is obtained from the TV array (θ-coordinates of blade at vertical mesh lines) for the blade surface involved. In the example, blade surfaces 1 and 2 pass through boundary 3. A portion of the TV array output from TANDEM for surfaces 1 and 2 is given in table VI. The m-coordinate corresponding to IM = 44 is circled along with the θ values, called TV, where the IM = 44 mesh line meets surfaces 1 and 2. The θ for blade surface 2 (or 4 on the rear blade of a tandem blade) must always have PITCH subtracted from it to bring it down on the same blade as surface 1 (or 3).
TABLE VI. - EXAMPLE OF TV ARRAY OUTPUT FROM TANDEM FOR BLADE SURFACES 1 AND 2

<table>
<thead>
<tr>
<th>m</th>
<th>BLADE SURFACE 1</th>
<th>BLADE SURFACE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TV</td>
<td>DTMV</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.10000E 11</td>
</tr>
<tr>
<td>0.27172E-02</td>
<td>-0.62034E-01</td>
<td>-69.7931</td>
</tr>
<tr>
<td>0.54344E-02</td>
<td>-0.24292</td>
<td>-63.4793</td>
</tr>
<tr>
<td>0.81517F-02</td>
<td>-0.40773</td>
<td>-57.9565</td>
</tr>
<tr>
<td>0.10869E-01</td>
<td>-0.54350</td>
<td>-53.2247</td>
</tr>
<tr>
<td>0.81702E-01</td>
<td>-2.37840</td>
<td>-11.8861</td>
</tr>
<tr>
<td>0.84683E-01</td>
<td>-2.41299</td>
<td>-11.3360</td>
</tr>
<tr>
<td>0.87664F-01</td>
<td>-2.44605</td>
<td>-10.8536</td>
</tr>
<tr>
<td>0.90645E-01</td>
<td>-2.47777</td>
<td>-10.4388</td>
</tr>
<tr>
<td>0.93626E-01</td>
<td>-2.50836</td>
<td>-10.0916</td>
</tr>
<tr>
<td>0.96607F-01</td>
<td>-2.53801</td>
<td>-9.81206</td>
</tr>
<tr>
<td>0.99588F-01</td>
<td>-2.56692</td>
<td>-9.60010</td>
</tr>
<tr>
<td>0.10257</td>
<td>-2.59531</td>
<td>-9.45574</td>
</tr>
<tr>
<td>0.10555</td>
<td>-2.62900</td>
<td>-0.10000E 11</td>
</tr>
</tbody>
</table>

All values of UBVIN are obtained from the table Stream-Function Values. A portion of that table for the example has been reproduced in table VII. The boundary values for the region of figure 11 have been circled in table VII, and the following paragraph explains how they were obtained.

The table of stream-function values gives u along vertical mesh lines from blade to blade. Each mesh line is listed separately, and if a second blade intersects the mesh line, the two parts of the mesh line are listed separately. On each part the IT of the first mesh point on the line above blade surface 1 or 3 is listed as IT1 in table VII. With this information, the proper boundary values for the region can be obtained.

Values in the shaded portion of the region of figure 11 must be obtained from the blade above, as shown in figure 10.

Along boundary 1 in the example it is desired to obtain u at IT = -97 for IM = 44 to 50. However, for IM = 44 to IM = 47, u must be obtained from the blade above the region in the figure (see fig. 10). Since NBBI (the number of horizontal mesh lines between AB and MN) is 28 in this example, u must be obtained at IT = -97 + 28 = -69 for these values of IM.

For IM = 44, ITL = -76 for the upper section of this vertical mesh line. Therefore, the eighth value in the row (0.79106) is the u for IT = -69. To reduce it to correspond to IT = -97, the stream-function period (1.0) is subtracted. The input value used is thus $0.79106 - 1.0000 = -0.20894$. Likewise, for IM = 45, IT1 = -77; and the ninth value (0.82429) corresponds to IT = -69. Reducing this by 1.0 gives -0.17571, the value used as input.
| IM = 42 | IT1 = -74 | 0.56370125 | 0.59389224 | 0.62004882 | 0.65130384 | 0.68482589 | 0.72076356 | 0.75924806 | 0.80028503 | 0.84364127 | 0.88884238 |
| IM = 43 | IT1 = -75 | 0.53544849 | 0.58359545 | 0.68012466 | 0.71162517 | 0.7459124 | 0.79547091 | 0.82392688 | 0.8639312 | 0.33717524 | 0.39157937 |
| IM = 44 | IT1 = -76 | 0.54564048 | 0.59254353 | 0.62132733 | 0.65190695 | 0.68458893 | 0.71922223 | 0.75647087 | 0.79649328 | 0.83945329 | 0.88507573 |
| IM = 45 | IT1 = -77 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 46 | IT1 = -78 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 47 | IT1 = -79 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 48 | IT1 = -107 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 49 | IT1 = -108 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 50 | IT1 = -109 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 51 | IT1 = -110 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |
| IM = 52 | IT1 = -111 | 0.56540487 | 0.59239887 | 0.62053421 | 0.65073729 | 0.68355896 | 0.71713170 | 0.75282977 | 0.79106661 | 0.83238629 | 0.87736242 |

TABLE VII. - STREAM-FUNCTION VALUE TABLE
After line 47, the region of figure 11 is normal, and values at \( IT = -97 \) are desired. Since \( IT1 = -107 \) for \( IM = 48 \) to 50, the 11th value in these rows corresponds to \( IT = -97 \).

Boundary 2 is easier for this example than boundary 1. At all values of \( IM \), \( u \) is desired for \( IT = -90 \). At \( IM = 44 \), \( IT = -90 \) corresponds to the first value in the row, which is 0.01517. For \( IM = 50 \), we need the 18th value \((-107 + 17 = -90)\), which is 0.21819.

On boundary 3, values of \( u \) are desired from mesh line 44. For the first section of the boundary, values must once again be obtained from the periodic blade above. \( IT = -97 \) to -93 is the desired range. Adding \( NBBI = 28 \) gives \( IT = -69 \) to -65. The values corresponding to these \( IT's \) are circled in table VII (0.79106 to 0.97554). Subtracting 1.0 from each of these gives -0.20894 to -0.02446. For the final point on the blade surface (point a), \( u = 0 \) is used.

The upper part of boundary 3 has two points. The first is again \( u = 0 \). The second, for \( IM = 44 \), is \( IT = -90 \). This point is the first stream-function value given for line 44, which is 0.01517.

Boundary 4 corresponds to \( IM = 50 \). Values are required from \( IT = -97 \) to \( IT = -90 \). Since \( IT1 = -107 \), the 11th to 18th values are desired. These are circled in table VII (-0.02076 to 0.21819).

After the final set of boundary values is given for boundary 4, a final data card must be given with zero for KBDRY and NSP. This signals the end of the data for MAGNFY.

**Output**

Generally, the MAGNFY output is similar to the 2DCP, TURBLE, or TANDEM output, but for the finer mesh. In MAGNFY the vertical mesh lines are numbered with \( IM = 1 \) for the left boundary to \( IM = MMM \) for the right boundary. The horizontal mesh lines are numbered with \( IT = 1 \) for the lower boundary to \( IT = ITMAX \) for the upper boundary.

Sample output is given in table VIII for the first example. Since the complete output would be lengthy and is similar to that for 2DCP, TURBLE, or TANDEM, the only output reproduced here is that which differs from the output of these three programs. The main part of table VIII is the stream-function values (UBV) and values of \( \rho \) times the component of \( \mathbf{W} \) normal to the boundary (RWBV) along the vertical and horizontal boundaries for the finer mesh. This main part is followed by a table of calculated program constants. The variable names are all defined in the section Main Dictionary.
TABLE VIII. - EXAMPLE OF MAGNIFY OUTPUT

STREAM FUNCTION AND RHO*W-SUB-THETA ON HORIZONTAL BOUNDARIES

<table>
<thead>
<tr>
<th>M</th>
<th>LOWER HORIZONTAL BOUNDARY</th>
<th>UPPER HORIZONTAL BOUNDARY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UBV</td>
<td>RwBV</td>
</tr>
<tr>
<td>0.96607E-01</td>
<td>-0.20894</td>
<td>-25.8414</td>
</tr>
<tr>
<td>0.97203E-01</td>
<td>-0.20220</td>
<td>-25.9841</td>
</tr>
<tr>
<td>0.97799E-01</td>
<td>-0.19550</td>
<td>-26.0991</td>
</tr>
<tr>
<td>0.98395E-01</td>
<td>-0.18884</td>
<td>-26.1856</td>
</tr>
<tr>
<td>0.98992E-01</td>
<td>-0.18225</td>
<td>-26.2423</td>
</tr>
</tbody>
</table>

STREAM FUNCTION AND RHO*W-SUB-M ON VERTICAL BOUNDARIES

<table>
<thead>
<tr>
<th>THETA</th>
<th>LEFT VERTICAL BOUNDARY</th>
<th>RIGHT VERTICAL BOUNDARY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UBV</td>
<td>RwBV</td>
</tr>
<tr>
<td>-2.72084</td>
<td>-0.20894</td>
<td>34.8187</td>
</tr>
<tr>
<td>-2.71523</td>
<td>-0.20090</td>
<td>35.1925</td>
</tr>
<tr>
<td>-2.70962</td>
<td>-0.19276</td>
<td>35.6345</td>
</tr>
<tr>
<td>-2.70401</td>
<td>-0.18451</td>
<td>36.1446</td>
</tr>
<tr>
<td>-2.69840</td>
<td>-0.17613</td>
<td>36.7226</td>
</tr>
</tbody>
</table>

CALCULATED PROGRAM CONSTANTS

<table>
<thead>
<tr>
<th>PITCH</th>
<th>HT</th>
<th>HM1</th>
<th>HM2</th>
<th>HM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7853982</td>
<td>0.5609987E-02</td>
<td>0.5434444E-03</td>
<td>0.5962105E-03</td>
<td>0.5962105E-03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MBII</th>
<th>MBD0</th>
<th>MMM</th>
<th>ITMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>-79</td>
<td>16</td>
<td>31</td>
<td>36</td>
</tr>
</tbody>
</table>

NUMBER OF INTERIOR MESH POINTS = 870

SURFACE BOUNDARY VALUES

<table>
<thead>
<tr>
<th>SURFACE</th>
<th>BV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>0.</td>
</tr>
<tr>
<td>3</td>
<td>-0.44431</td>
</tr>
<tr>
<td>4</td>
<td>-0.44431</td>
</tr>
</tbody>
</table>

Error Conditions

The error conditions are as follows:

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

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

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

(3) ROOT CALL NO. XX

ROOT HAS FAILED TO CONVERGE IN 1000 ITERATIONS
This message is printed by subroutine ROOT if a root cannot be located. The input to ROOT is also printed. The user should thoroughly check the input to the main program.

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

(4) DENSTY CALL NO. XX

\[ \text{NER}(1) = \text{XX} \]

\( \text{RHO}^*\text{W} \) IS \( \text{X.XXXX} \) TIMES THE MAXIMUM VALUE FOR \( \text{RHO}^*\text{W} \)
This message is printed if the value of \( \rho W \) at some mesh point is so large that there is no solution for the values of \( \rho \) and \( W \). This indicates a locally supersonic condition, which can be eliminated by decreasing \( \text{WTFL} \) in the original 2DCP, TURBLE, or TANDEM run to obtain new input boundary values for MAGNFY.

If \( \text{RHO}^*\text{W} \) is too large, MAGNFY still attempts to calculate a solution. This often permits an approximate solution to be obtained which is valid at all the subsonic points in the region. In other cases, the value of \( W \) is reduced at some of the points in question during later iterations, resulting in a valid final solution for these points. The program counts the number of times supersonic flow has been located at any point during a given run (NER(1)). When NER(1) = 50, the program is stopped.

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

(5) THE USER HAS FAILED TO SPECIFY WHICH TYPE OF INPUT HE IS USING
The first card of input after the title card specifies whether a single or tandem blade is being considered. There must be a 1 or a 2 in column 5 of this card.

(6) MMM GT 100, OR ITMAX GT 50
This is printed if MMM > 100 or if ITMAX > 50. In this case either MAGFAC should be reduced, or a smaller region chosen.

(7) ONE OF THE MH ARRAYS IS TOO LARGE
This is printed if there are more than 100 intersections of horizontal mesh lines with any blade surface. In this case MAGFAC should be reduced, or a smaller region chosen.
THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2000

This is printed if there are more than the allowable number of finite-difference grid points. Either MAGFAC must be reduced, or a smaller region must be chosen.

SEARCH CANNOT FIND M IN THE MH ARRAY.

If this is printed, the value of m and the blade-surface number are also printed. The user should thoroughly check the input to the main program.

PROGRAM PROCEDURE

The program is segmented into seven main parts - the subroutines INPUT, PRECAL, COEF, SOR, SLAX, TANG, and VELOCY, called by the main program MAGFNY. In addition there are several other subroutines. All the subroutines and their relation are shown in figure 12. All information which must be transmitted between the seven main subroutines is placed in COMMON.

Although most subroutines have been changed from those in TANDEM (ref. 3), the subroutine names have not been changed. Even with the rather extensive revisions made in some subroutines, the general descriptions of the subroutines in reference 3 still apply with minor differences, except for INPUT and PRECAL. Therefore, the only subroutines described here are INPUT and PRECAL, plus two new subroutines, WRITU and BDVINT.

Most of the subroutines in MAGFNY use the same set of variables. And most of these variables are the same as those used in TANDEM. These variables are all defined in the Main Dictionary (p. 32). The subroutines using these variables are described prior to the Main Dictionary or in reference 3. The remaining subroutines are described after
the Main Dictionary or in reference 3, and variables are defined with each subroutine.

The program can handle as many as 2000 mesh points on the IBM 2-7094-7044 direct-coupled system with a 32 768-word core. To provide for the handling of 2000 mesh points an overlay arrangement is used, as shown in figure 13. All subroutines not shown are in the main link. The total program storage requirement is 74044(8) of which 53364(8) is in COMMON blocks which are stored in the main link. The system storage requirement for our computer is 2764(8) and unused storage is 750(8). If there is a storage problem on the user's computer, the maximum number of mesh points should be reduced. The following program changes are required to change the maximum number of mesh points:

(1) Change the dimension of A, U, K, and RHO in the COMMON/AUKRHO statement to the maximum allowable number of mesh points. This statement occurs in most subroutines.

(2) In subroutine INPUT change the number of values of K and RHO to be initialized (the bound on the DO loop near statement 240).

(3) In subroutine PRECAL change the statement following statement 210 and format statement 1130 to reflect the maximum allowable number of mesh points. The statement following statement 210 will cause the program to stop if there are too many mesh points.

(4) Change the dimensions of W, RWM, and BETA in SLAX, SLAVBB, TANG, VELOCY, and VELBB.

(5) If the number of mesh points is reduced below 1600, the EQUIVALENCE state-
ments in SLAX, SLAVBB, TANG, VELOCY, and VELBB must be changed.

The first segment of the program is INPUT. This subroutine reads all input data cards, calculates constants, and initializes arrays. It also uses SPLINE interpolation on the input boundary values to obtain boundary values on the fine mesh. The next subroutine is PRECAL, which calculates all quantities that remain constant for a single problem. INPUT and PRECAL are each called once for a given problem. The remaining subroutines are each called once for each outer iteration. The subroutine COEF calculates the entries of the matrix $A$ and the vector $k$ of equation (A7) (ref. 1). These coefficients must be recalculated for each outer iteration. On the first outer iteration subroutine SOR estimates an optimum overrelaxation parameter $\omega$ on the first call if it is not given as input. The same value of $\omega$ is used for each outer iteration. SOR then finds the linear solution to equation (A7) (ref. 1) with fixed coefficients by successive overrelaxation. Then subroutine SLAX calculates $\rho W_m$. Subroutine TANG calculates $\rho W_\theta$, and then $\rho W$ and $\beta$ throughout the region. Finally, the subroutine VELOCY calculates density $\rho$ and velocity $W$ throughout the region and on the blade surfaces, and plots the surface velocities.

Conventions Used in Program

In general, the same conventions are used in MAGNFY as were used in the TANDEM program (ref. 3). In addition the lower, upper, left, and right boundaries of the magnified region are numbered 1, 2, 3, and 4, respectively. Also, the lower and upper boundaries of the region must sometimes be considered as blade surfaces by the program. In these cases they are numbered 5 and 6, respectively.

Labeled COMMON Blocks

The labeled COMMON blocks are organized the same as for TANDEM (ref. 3), except for the omission of /SLA/.

Subroutine INPUT

Read and print first part of input. - The program first reads the input cards which are the same as those for TURBLE or TANDEM, or were modified from 2DCP. A description of this input is given in reference 1, 2, or 3. All the input data are printed as they are read in.

Fill in dummy second blade for single-blade case. - When there is only one blade,
the arrays for the second blade (surfaces 3 and 4) are filled in with the data for the first blade.

Calculate large mesh arrays. - The large mesh spacing and MV array are calculated first so that the RMI and RMO arrays can be calculated.

Transfer blade coordinates for tandem blade when necessary. - If a leading-edge region is being analyzed, it is always considered to be on the rear blade. Hence, if it is the leading edge of the first blade of a tandem or slotted blade, the arrays for the rear blade are filled with the data for the front blade. Similar considerations hold for a trailing-edge region, since it is always considered to be on the front blade.

Calculate constants. - Geometrical and miscellaneous constants are calculated.

Calculate fine-mesh m-coordinates. - The final MV array for the fine mesh can now be calculated.

Read boundary values and interpolate for fine mesh. - Each set of coarse-mesh boundary value data is read. Then interpolated stream-function values for the fine mesh are calculated and printed.

Subroutine PRECAL

The calculation of \( \lambda \) and other constants in PRECAL is no longer necessary in MAGIFY, since \( \lambda \) is given as input and the other constants are not used in the calculation. The remainder of the description of PRECAL in reference 3 is still valid.

Subroutine WRITU

WRITU prints the value of the stream function along a given vertical line between blades or boundaries. A label is printed with the value of IM for the mesh line and the value of IT(IT1) for the first printed stream-function value.

Main Dictionary

The Main Dictionary applies to the previously discussed subroutines.

A array of coefficients of \( u \) (i.e., elements of \( a_{ij} \) of matrix A in eq. (A7) of ref. 1)
A12, A34  \( a_{12}, a_{34} \) in eq. (A2) of ref. 1

AA  temporary variable in BLCD

AAA  array used for temporary storage

AANDK  input variable

AATEMP  temporary location for AANDK in SOR

ANS  result of calls on ROOT in TANG and DENSTY in SLAVBB and VELBB

AR  input variable (from 2DCP, TURBLE, or TANDEM)

AZ  \( a_0 \) in eq. (A2) of ref. 1

B  array containing stream-channel thickness \( b \) at the four points adjacent to a point for which AAK is called

B12, B34  \( b_{12}, b_{34} \) in eq. (A2) of ref. 1

BB  temporary variable in BLCD

BE  array of values of \( b \) at vertical mesh lines

BEH  array of values of \( b \) where horizontal mesh lines meet the four blade surfaces

BESP  input variable (from 2DCP, TURBLE, or TANDEM)

BETA  array of values of \( \beta \) at interior mesh points

BETAH(BETAV)  array of values of \( \beta \) where horizontal (vertical) mesh lines meet the four blade surfaces

BETA1(BETAO)  input variable (from 2DCP, TURBLE, or TANDEM)

BETI(BETO)  array of angles at tangent points of leading- (trailing-) edge radii with the four blade surfaces (see input BETI1, 2, 3, 4 and BETO1, 2, 3, 4)

BLDAT  input variable

BV  array of stream-function boundary values on the four blade surfaces

BVIN  input variable

BZ  stream-channel thickness \( b_0 \) at point for which AAK is called
CDMBIT(CDMBOT) temporary grid locations along meridional axis in INPUT
CHANGE change in value of stream function at a particular point during an iteration of SOR
CHORD array containing the meridional chord distances of each of the four blade surfaces (see input CHORDF and CHORDR)
CMM temporary variable in BLCD
CP specific heat at constant pressure, \( c_p \)
CPTIP \( 2c_pT_{in} \)
DBDM array of slopes at vertical mesh lines of spline curve for stream-channel thickness
DIST meridional distance in SEARCH from a blade leading edge to where a horizontal mesh line meets a blade surface
DMLR tolerance for mesh points near a boundary in the m-direction (If a mesh point is closer than DMLR to a boundary, the point is considered to be on the boundary.)
DTDM \( d\theta/dm \) along a blade surface in BLCD
DTDMH(DTDMV) array of \( d\theta/dm \) where horizontal (vertical) mesh lines meet the four blade surfaces
DTLR tolerance in \( \theta \)-direction (see DMLR)
DUDM array of derivatives of stream function \( \partial u/\partial m \) along horizontal mesh lines in meridional direction
DUDT array of derivatives of stream function \( \partial u/\partial \theta \) along vertical mesh lines in \( \theta \)-direction
EM array of second derivatives of spline curves for each blade surface, calculated by SPLN22 in BLCD
EMK, EMKM1 temporary variables for EM in BLCD
ERROR maximum absolute value of change in \( u \) at any point for an over-relaxation (SOR) iteration
ERSOR input variable
EXPON \( 1/(\gamma - 1) \)
FIRST initial value of some index
GAM input variable (from 2DCP, TURBLE, or TANDEM)
H array containing mesh spacing $h$ between the point for which AAK is called and the four points adjacent to it

HM1 mesh spacing in m-direction from upstream boundary through front blade

HM2 mesh spacing in m-direction for overlapping portion of front and rear blades, or between blades for nonoverlapping case

HM3 mesh spacing in m-direction through rear blade to downstream boundary

HT mesh spacing in $\theta$-direction from blade to blade

I temporary integer variable in INPUT, PRECAL, SLAX, and SEARCH

IEND integer variable set equal to 1 when final convergence to a solution is reached in the outer iterations on a given set of data

IH array containing current number of intersections of horizontal mesh lines with each of the four blade surfaces as intersections are located

IHS integer variable in BDRY34 and TANG for counting intersections of horizontal mesh lines with blade surfaces

IM index of mesh line in the meridional direction (m-direction)

IM1(IMT) integer variable in TANG indicating vertical mesh line index of the first (final) point in region of a horizontal mesh line

IM2 IM1 + 1

IMS array containing total number of intersections of horizontal mesh lines with each of the four blade surfaces

IMSL temporary variable in PRECAL

IMSS temporary variable in PRECAL, VELOCY, and VELBB

IMTM1 IMT - 1

INF variable in PRECAL indicating when an infinite slope is located at a blade leading or trailing edge in a call on BLCD

INIT array used to indicate whether BLCD has been called previously on a given blade surface

INTVL input variable

IP index of mesh point

34
IP1, IP2, IP3, IP4
value of IP at the four points adjacent to the mesh point under consideration

IPL(IPU)
value of IP where a vertical mesh line meets a lower (upper) surface or boundary

IPLM1(IPUP1)
value of IP on a vertical mesh line adjacent to a lower (upper) surface in VELBB

IS
integer variable in SEARCH for indicating where a horizontal mesh line intersects a blade surface

IT
index of mesh line in $\theta$-direction

ITF(ITL)
input variable

ITER
outer iteration counter

ITMAX
maximum value of IT in magnified mesh region

ITOR
value of IT at origin of coordinates at leading edge of front blade

ITV
array of horizontal mesh line indexes (IT) corresponding to intersections of vertical mesh lines with blade surfaces (ITV(IM, SURF) is the IT value for the mesh point in the region on vertical mesh line IM which is closest to blade surface SURF. If ITV $\leq$ 1, the value is adjusted to 1 for a lower surface or 2 for an upper surface. If ITV $\geq$ ITMAX, the value is adjusted to ITMAX - 1 for a lower surface or ITMAX for an upper surface. If a vertical line does not intersect a blade surface, its value of ITV(IM, SURF) is equal to -10000. For the lower boundary (SURF = 5), ITV = 2; and for the upper boundary (SURF = 6), ITV = ITMAX - 1.)

ITVL(ITVU)
ITV of lower (upper) blade surface on a given vertical mesh line

ITVM1(ITVP1)
ITV of a blade surface in COEFBB for vertical mesh line to left (right) of the line under consideration

IV
array containing the value of IP at the base of each vertical mesh line

J
temporary integer variable in INPUT

K
array of constants; the vector $k$ in eq. (A7) of ref. 1

KA
integer array indicating which of the four points surrounding a mesh point lie on a boundary
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAK</td>
<td>real array giving boundary values of points surrounding a mesh point next to a boundary</td>
</tr>
<tr>
<td>KBDRY</td>
<td>input variable</td>
</tr>
<tr>
<td>KK</td>
<td>integer counter in BLCD</td>
</tr>
<tr>
<td>KKK</td>
<td>array containing information used in plotting subroutine PLOTMY</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>input variable</td>
</tr>
<tr>
<td>LAST</td>
<td>final value of some index</td>
</tr>
<tr>
<td>LER</td>
<td>array indicating location of error messages printed by program</td>
</tr>
<tr>
<td>LMAX</td>
<td>maximum value of $u_1^{m+1}/u_1^m$ for eq. (B2) of ref. 7</td>
</tr>
<tr>
<td>LOWER</td>
<td>integer variable representing one of lower blade surfaces, 2 or 4</td>
</tr>
<tr>
<td>M</td>
<td>meridional coordinate, meters</td>
</tr>
<tr>
<td>MAGFAC</td>
<td>input variable</td>
</tr>
<tr>
<td>MB1, MB2</td>
<td>temporary vertical grid line locations along meridional axis</td>
</tr>
<tr>
<td>MBDYF, MBDYL</td>
<td>input variable</td>
</tr>
<tr>
<td>MBI</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>MBI2</td>
<td>input variable (from TANDEM)</td>
</tr>
<tr>
<td>MBII</td>
<td>number of vertical mesh lines from left boundary to leading edge of a blade in the magnified region (If region surrounds trailing edge of rear blade of a tandem blade, MBII = 1000.)</td>
</tr>
<tr>
<td>MBIIM1</td>
<td>MBII - 1</td>
</tr>
<tr>
<td>MBIIP1</td>
<td>MBII + 1</td>
</tr>
<tr>
<td>MBIT, MBOT</td>
<td>temporary grid locations along meridional axis</td>
</tr>
<tr>
<td>MBO</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>MBO2</td>
<td>input variable (from TANDEM)</td>
</tr>
<tr>
<td>MBOO</td>
<td>number of vertical lines from left boundary to trailing edge of a blade in the magnified region (If region surrounds leading edge of front blade of a tandem blade, MBOO = -1000.)</td>
</tr>
<tr>
<td>MBOOM1</td>
<td>MBOO - 1</td>
</tr>
<tr>
<td>MBOOP1</td>
<td>MBOO + 1</td>
</tr>
<tr>
<td>MH</td>
<td>array of $m$-coordinates of intersections of horizontal mesh lines with the four blade surfaces</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MLE</td>
<td>array of m-coordinates of leading edges of the four blade surfaces (see input MLE2)</td>
</tr>
<tr>
<td>MM</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>MMM</td>
<td>number of vertical mesh lines from the left to right boundaries of the magnified region</td>
</tr>
<tr>
<td>MMMM1</td>
<td>MMM - 1</td>
</tr>
<tr>
<td>MMLE</td>
<td>temporary meridional distance in BLCD</td>
</tr>
<tr>
<td>MMMSP</td>
<td>temporary meridional distance in BLCD</td>
</tr>
<tr>
<td>MR</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>MRTS</td>
<td>integer switch in PRECAL indicating when infinite derivatives would be encountered in a call on MHORIZ</td>
</tr>
<tr>
<td>MSP</td>
<td>array of m-coordinates of spline points for each blade surface measured from its leading edge (see input MSP1, 2, 3, 4)</td>
</tr>
<tr>
<td>MSPMM</td>
<td>temporary meridional distance in BLCD</td>
</tr>
<tr>
<td>MV</td>
<td>array of m-coordinates of vertical mesh lines</td>
</tr>
<tr>
<td>NBBI</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>NBL</td>
<td>number of blades</td>
</tr>
<tr>
<td>NER</td>
<td>array indicating number of times certain error messages are printed by program</td>
</tr>
<tr>
<td>NIP</td>
<td>number of interior mesh points</td>
</tr>
<tr>
<td>NOBL</td>
<td>input variable</td>
</tr>
<tr>
<td>NP1, NP2</td>
<td>integer counters in VELOCY indicating number of plotted blade-surface velocities</td>
</tr>
<tr>
<td>NRSP</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>NSP</td>
<td>input variable</td>
</tr>
<tr>
<td>NSPI</td>
<td>array containing number of spline points on each of the four blade surfaces (see input SPLNO1, 2, 3, 4)</td>
</tr>
<tr>
<td>NSPM1</td>
<td>NSP - 1</td>
</tr>
<tr>
<td>OMEGA</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>ORF</td>
<td>input variable (from 2DCP, TURBLE, or TANDEM)</td>
</tr>
<tr>
<td>ORFOPT</td>
<td>upper bound for optimum $\Omega$ from eqs. (B1) and (B2) of ref. 7</td>
</tr>
</tbody>
</table>
ORFTEM  temporary storage for ORFOPT
P  array containing information used in the plotting subroutine, PLOTMY
PITCH  $\theta$-coordinate from blade to blade, $2\pi/NBL$
R  array of densities $\rho$ at the four points adjacent to a point for which AAK is called
RATIO  value of $u_{1}^{m+1}/u_{1}^{m}$ for use in eqs. (B2) and (B3) of ref. 7
RBV  array of densities $\rho$ on the four boundaries of the magnified region
RELER  maximum relative change in density at surface mesh points between two outer iterations
RHO  array of densities $\rho$ at interior mesh points
RHOB  temporary storage in VELBB for a value of $\rho$ on a blade surface
RHOHB  array of densities $\rho$ at horizontal mesh line intersections with the four blade surfaces
RHOIP  input variable (from 2DCP, TURBLE, or TANDEM)
RHOVB  array of densities $\rho$ at vertical mesh line intersections with the four blade surfaces
RI(RO)  array of leading- (trailing-) edge radii on the four blade surfaces (see input RI1, 2, 3, 4 and RO1, 2, 3, 4)
RM  array of $r$-coordinates of the mean stream surface radii at vertical mesh lines
RMDTL2(RMDTU2)  $[r(\partial\theta/\partial m)]^2$ at vertical mesh line intersections on lower (upper) blade surfaces
RMH  array of $r$-coordinates of the stream surface radii where horizontal mesh lines meet the four blade surfaces
RMI(RMO)  array of $r$-coordinates of mean stream surface radii at the inlet (outlet) of the four blade surfaces
RMM  temporary meridional distance in BLCD
RMSP  input variable (from 2DCP, TURBLE, or TANDEM)
RW  value of $\rho W$ of a mesh point
RWBV  array of $\rho$ times the velocity component normal to the boundary of the magnified region
RWM  
array of \rho W_m \text{ where vertical mesh lines intersect the four blade surfaces}

RWT  
array of \rho W_\theta \text{ where horizontal mesh lines intersect the four blade surfaces}

RWMBV  
value of \rho W_m \text{ at a mesh point along the lower or upper boundary of the magnified region}

RZ  
s\text{ density } \rho_0 \text{ at point for which AAK is called}

S  
meridional distance between two adjacent blade-surface spline points in BLCD

S1(ST)  
blade-surface number at the beginning (end) of a horizontal mesh line in TANG

SAL  
array of values of \sin \alpha = dr/dm \text{ at each vertical mesh line}

SIGN  
integer constant in BLCD

SPLNO  
number of input spline points on a blade surface

SPM  
array of m-coordinates along a horizontal mesh line in TANG

SRW  
integer code variable that will cause certain subroutines to write out useful data for debugging:

SRW = 13, SPLINE will write input and output data.
SRW = 16, SPLINT will write input and output data.
SRW = 18, SPLN22 will write input and output data.
SRW = 21, ROOT will write input and successive estimates of the root to which it is converging.

STGR  
array of \theta-coordinates of center of each trailing-edge radius with respect to center of its leading-edge radius (see input STGRF and STGRR)

STRFN  
input variable

SURF  
integer variable referring to each of the four blade surfaces, the lower boundary (5), or the upper boundary (6)

SURVL  
input variable

T1, T2  
elapsed time in clock pulses (1/60 sec)

TBI  
\tan \beta_{le}

TBO  
\tan \beta_{te}

TGROG  
2\gamma R/(\gamma + 1)
array in INPUT; also single variable in PRECAL and TANG containing θ-coordinates from leading edge of front blade to a horizontal mesh line

θ-coordinate of a point along a blade surface in BLCD

temporary variables in BLCD

array of θ-coordinates from origin of front blade to leading edge of each blade surface (see input THLE2)

array of θ-coordinates of spline points for each blade surface measured from its leading edge (see input, THSP1, 2, 3, 4)

elapsed time in minutes

input variable (from 2DCP, TURBLE, or TANDEM)

array of θ-coordinates of points along a vertical mesh line in SLAVBB

array of θ-coordinates where vertical mesh lines meet the four blade surfaces

2ωλ

2ωλ - (ωr)^2

2ω/w

array of stream-function values at each mesh point, or of eigenvector associated with calculation of ORFOPT

array of values of stream function u at mesh points on boundary of magnified region

input variable

new value of stream-function estimate at a single point calculated by eq. (7) of ref. 1

integer variable representing one of the upper blade surfaces, 1 or 3

array of values of stream function along a vertical or horizontal mesh line, including boundary points

array of relative velocities W at unknown mesh points, also used for storing ρW

critical velocity on a blade surface
array of \( \rho W_m \) where vertical mesh lines intersect the four blade surfaces

array of \( \rho W_\theta \) where horizontal mesh lines intersect the four blade surfaces

input variable (from 2DCP, TURBLE, or TANDEM)

input variable (from TANDEM)

array, ratio of blade-surface velocity (based on meridional components) to critical velocity

array, ratio of blade-surface velocity (based on tangential components) to critical velocity

array of \( m \)-coordinates where surface velocities are plotted

array of surface velocities to be plotted

Program Listing for Subroutines Using Main Dictionary

```fortran
COMMON SRW, ITER, IEND, LER(2), NER(1)
COMMON /INP/ GAM, AR, TIP, RHOIP, TFL, WTFLSP, UMFGA, ORF, RETAI, BETA(I),
1 NOBL, MBI, MBU, MBZ, MB2, MM, NBI, NOB, NRSP, MBDYF, MBDYL, ITF, ITL,
2 HLDT, AANDK, ERSOR, STRFN, INTVL, SURVL, MAGFAC,
3 MR(J0), RMSP(50), BESP(50)
COMMON /CALLUN/ MBI, MBUD, MM, MBIM1, MBIPL, MBUDM1, MBUP1, MMB1,
1 HML, HM2, HM3, HT, DTLR, DMLR, PITCH, CP, EXPON, TWW, CPTIP, TGRUO, TB1, TB2,
2 LAM, DA, TWL, ITUR, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3 UBV(100,4), KBDV(100,4), IV(100,6), IV(100,4), UTDMV(100,4),
4 BETV(100,4), NMV(100,4), UTDMVH(100,4), BETAH(100,4), KM(100,4),
5 BEMH(100,4), RHM(100,4), RBE(100), DBNVH(100), SAL(100), AAA(100)
COMMON /GFMIN/ CHORD(4), STGR(4), MLE(4), THLE(4), RMI(4), RMU(4),
1 R1(4), RD(4), OLT(4), RTO(4), NSPI(4), MSP(50,4), IHSP(50,4)
COMMON /RHDV/ RHDVH(100,4), RHDV(100,4), RV(100,4),
1 COMMON /BLCCCM/ EM(50,4), INIT(4)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AAITEM, SURF,
1FIRST, UPPPER, SL, ST, SRW
REAL K, KAK, LAMBD, LMAX, MH, MLF, MR, MSP, MV, MVIM1
CALL TIME1(11)
10 IEND = -1
ITER = 0
DO 20 SURF=1,4
20 INIT(SURF) = 0
CALL INPUT
CALL PRECAL
30 CALL COEF
CALL SOR
CALL TIME1(T2)
TIME = (T2-T1)/3600.
WRITE(6,1000) TIME
```

41
CALL SLAX
CALL TANG
CALL VELOCY
CALL TIME(T2)
TIME = (T2 - T1) / 3600.
WRITE(6,1000) TIME
IF (IEND) 30, 10

1000 FORMAT (8HLTITIME = ,F7.4, 5H MIN.)
END

SUBROUTINE INPUT

C INPUT READS AND PRINTS ALL INPUT DATA CARDS AND CALCULATES HORIZONTAL MESH SPACING (MV ARRAY)

COMMON SRW, ITER, IEND, LER(2), NER(1)
COMMON /INP/ GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, ORF, BETAI, BETAO,
1 NDBL, MBL, MB1, MB2, MB3, MB4, MB5, MB6, MB7, MB8, MB9, MB10,
2 SPLAT, AANDK, ERSOR, STRFN, INTVL, SURVL, MAGFA, 3MR(50), RMS(50), BESP(50)
COMMON /CALCON/ MBII, MB30, MMB, MBB, MBB2, MB2, MB3, MB4, MB5, MB6, MB7, MB8,
1 HMLB, HML2, HML3, HML4, DTLR, DMLR, PITCH, CP, EXPON, TWW, CPTIP, TGRGO, TBI, TBO,
2 LAMBDA, TML, ITOR, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3 VBV(100,4), KBVBV(100,4), ITV(100,4), TV(100,4), DTMV(100,4),
4 HET(100,4), MH(100,4), DTXMH(100,4), BETAH(100,4), RMH(100,4),
5 REH(100,4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)
COMMON /GEOMIN/ CHORD(4), STGR(4), MLE(4), THLE(4), RMI(4), RMO(4),
1 R(4), RO(4), BETI(4), BETO(4), NSPI(4), MSP(50,4), IHSP(50,4)
COMMON /RHOS/ RHOHB(100,4), RHOBV(100,4), RVB(100,4)
DIMENSION BVIN(100), UBVIN(100), TH(100)
INTEGER BLAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1 FIRST, UPPER, ST, SRW
REAL X, KX, LAM, LMAX, M, MLE, MR, MSP, MV, MVIM

C READ AND WRITE INPUT DATA

WRITE(6,1000)
READ (5,1100)
WRITE(6,1100)
READ (5,1010) NDBL
WRITE(6,1110) NDBL
IF (NDBL.EQ.1.OR.NDBL.EQ.2) GO TO 10
WRITE (6,1120)
STOP

C OLD TURBLE (2) OR TANDEM DATA
10 IF (NDBL.EQ.1) WRITE(6,1130)
    IF (NDBL.EQ.2) WRITE(6,1140)
    READ (5,1030) GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, ORF
    WRITE(6,1040) GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, ORF
    IF (NDBL.EQ.1) WRITE(6,1150)
    IF (NDBL.EQ.2) WRITE(6,1150)
    READ(5,1030) BETAI, BETAO, CHORD(1), STGR(1), CHORD(3), STGR(3),
    MLE(3), THLE(3)
    WRITE(6,1040) BETAI, BETAO, CHORD(1), STGR(1), CHORD(3), STGR(3),

42
IMLE(3), THLE(3)
IF (NOBL.EQ.1) WRITE(6,1170)
IF (NOBL.EQ.2) WRITE(6,1180)
READ (5,1010) MBI, MB0, MB12, MB02, MM, WBBI, NBL, NRSP
WRITE(6,1010) MBI, MB0, MB12, MB02, MM, WBBI, NBL, NRSP
DO 20 J=1,4
IF (NOBL.EQ.1.AND.J.EQ.3) GO TO 30
IF (J.EQ.1) WRITE(6,1190)
IF (J.EQ.2) WRITE(6,1200)
IF (J.EQ.3) WRITE(6,1210)
IF (J.EQ.4) WRITE(6,1220)
WRITE(6,1230) J, JJ, J, J
READ (5,1030) RI(J), RO(J), BETI(J), BETO(J), SPLNU
WRITE(6,1040) RI(J), RO(J), BETI(J), BETO(J), SPLNU
NSPI(J) = SPLNU
NSP = NSPI(J)
WRITE(6,1240) J
READ (5,1030) (MSP(I,J), I=1, NSP)
WRITE(6,1040) (MSP(I,J), I=1, NSP)
WRITE(6,1250) J
READ (5,1030) (THSP(I,J), I=1, NSP)
20 WRITE(6,1040) (THSP(I,J), I=1, NSP)
30 WRITE(6,1260)
READ (5,1030) (MR(I), I=1, NRSP)
WRITE(6,1040) (MR(I), I=1, NRSP)
WRITE(6,1270)
READ (5,1030) (RMSP(I), I=1, NRSP)
WRITE(6,1040) (RMSP(I), I=1, NRSP)
WRITE(6,1280)
READ (5,1030) (BESP(I), I=1, NRSP)
WRITE(6,1040) (BESP(I), I=1, NRSP)
C NEW MAGNIFY DATA
WRITE(6,1290)
READ (5,1010) BLDAT, AANDK, ERSOR, STRFN, INTVL, SURVL
WRITE(6,1020) BLDAT, AANDK, ERSOR, STRFN, INTVL, SURVL
WRITE(6,1300)
READ (5,1010) MBDFY, MBDFY, ITF, ITL, MACFAC
WRITE(6,1020) MBDFY, MBDFY, ITF, ITL, MACFAC
WRITE(6,1310)
READ (5,1030) LAMBDA
WRITE(6,1040) LAMBDA
MLE(1) = 0.
THLE(1) = 0.
C FOR SINGLE BLADE CASE, FILL IN DUMMY TANDEM BLADE
C
IF (NOBL.EQ.2) GO TO 60
WTFLSP = 0.
CHORD(3) = CHORD(1)
STGR(3) = STGR(1)
MLE(3) = 0.
THLE(3) = 0.
MBI2 = MBI
MB02 = MB0
RI(3) = RI(1)
RI(4) = RI(2)
RO(3) = RO(1)
RO(4) = RO(2)
BETI(3) = BETI(1)
CALCULATE LARGE MESH SPACING

60 HM1 = CHORD(1)/FLOAT(MBO-MBI)
    IF (MBO.GT.MBI2 .AND. MB1.NE.MBI2) HM1 = MLE(3)/FLOAT(MBI2-MBI1)
    HM2 = 1.E30
    IF (MBI2.NE.MBO) HM2 = (CHORD(1)-MLE(3))/FLOAT(MBO-MBI2)
    HM3 = CHORD(3)/FLOAT(MBO2-MBI2)
    IF (MBO.GT.MBI2 .AND. MB0.NE.MBI2) HM3 = (CHORD(3)+MLE(3)-CHORD(1))
1/FLOAT(MBO2-MBO)
    PITCH = 2.*3.1415927/FLOAT(NBM1)
    HT = PITCH/FLOAT(NBM1)

CALCULATE LARGE MESH MV ARRAY, AND RMI,RMO, AND BV ARRAYS

MBII = MINO(MBO,MBI2)
CDMBOT = AMIN1(CHORD(1),MLE(3))
DO 70 IM=1,MBII
70 MV(IM) = FLOAT(IM-MBI)*HM1
    MBIT = MAX0(MBU,MBI2)
    CDMBIT = AMAX1(CHORD(1),MLE(3))
    DO 80 IM=MBIT,MBIT
80 MV(IM) = CDMBIT+FLOAT(IM-MBIT)*HM2
    DO 90 IM=MBIT,MM
90 MV(IM) = CDMBIT+FLOAT(IM-MBIT)*HM3
    CALL SPLINT (MR,NMSP,NRSP,MV,MM,MR,SM)
    RMI(1) = RM(MBI)
    RMI(2) = RM(MBI)
    RMI(3) = RM(MBI2)
    RMI(4) = RM(MBI2)
    RMO(1) = RM(MBO)
    RMO(2) = RM(MBO)
    RMO(3) = RM(MBO2)
    RMO(4) = RM(MBO2)
    BVV(1) = 0.
    BVV(2) = 0.
    BVV(3) = -WTFLSP/WTFL
    BVV(4) = BVV(3)

CALCULATE GEOMETRICAL CONSTANTS

MBII = (MBI2-MBDYF)*MAGFAC+1
MB00 = (MBO-MBDYF)*MAGFAC+1
MM = (MBDYL-MBDYF)*MAGFAC+1
ITMAX = (ITL-ITF)*MAGFAC+1
HM1 = HM1/FLOAT(MAGFAC)
HM2 = HM2/FLOAT(MAGFAC)
HM3 = HM3/FLOAT(MAGFAC)
HT = HT/FLOAT(MAGFAC)
ITOK = -ITF*MAGFAC+1
IF (NOBL.EQ.1) GO TO 130
C
C FOR TANDEM BLADE CASE, IF REGION SURROUNDS LEADING EDGE OF
C FRONT BLADE, STORE BLADE SURFACES 1 AND 2 INTO 3 AND 4
C
IF (MBDYF.GE.MBI.OR.MBDYL.LE.MBI) GO TO 110
IF (ITF.GE.0.OR.ITL.LE.0) GO TO 110
MBII = MBI-MBDYF*MAGFAC+1
MBOJ = -1000
HM2 = HM1
HM3 = HM1
CHORD(3) = CHORD(1)
STGK(3) = STGK(1)
MLE(3) = MLE(1)
THLE(3) = THLE(1)
DO 100 J=1,3
RI(J+2) = RI(J)
RO(J+2) = RO(J)
BETI(J+2) = BETI(J)
BETO(J+2) = BETO(J)
RMI(J+2) = RMI(J)
RMO(J+2) = RMO(J)
BV(J+2) = BV(J)
NSPI(J+2) = NSPI(J)
NSP = NSPI(J)
DO 100 I=1,NSP
MSP(I,J+2) = MSP(I,J)
100 THSP(I,J+2) = THSP(I,J)
C
C FOR TANDEM BLADE CASE, IF REGION SURROUNDS TRAILING EDGE OF
C FAR BLADE, STORE BLADE SURFACES 3 AND 4 INTO 1 AND 2
C
110 IF (MBDYF.GE.MBO2.OR.MBUYL.LE.MBO2) GO TO 130
IF (FLOAT(ITF*MAGFAC)*HT.GE.THLE(3)+STGK(3).OR.FLOAT(ITL*MAGFAC)*
HT.LE.THLE(3)+STGK(3)) GO TO 130
MBII = 1000
MBOJ = (MBO2-MBDYF)*MAGFAC+1
HM1 = HM3
HM2 = HM3
CHORD(1) = CHORD(3)
STGK(1) = STGK(3)
MLE(1) = MLE(3)
THLE(1) = THLE(3)
DO 120 J=3,4
RI(J-2) = RI(J)
RO(J-2) = RO(J)
BETI(J-2) = BETI(J)
BETO(J-2) = BETO(J)
RMI(J-2) = RMI(J)
RMO(J-2) = RMO(J)
BV(J-2) = BV(J)
NSPI(J-2) = NSP(J)
NSP = NSPI(J)
DO 120 I=1,NSP
   MSP(I,J-2) = MSP(I,J)
120   THSP(I,J-2) = THSP(I,J)

CALCULATE MISCELLANEOUS CONSTANTS

CHORD(2) = CHORD(1)
CHORD(4) = CHORD(3)
STGR(2) = STGR(1)
STGR(4) = STGR(3)
MLE(2) = MLE(1)
MLE(4) = MLE(3)
THLE(2) = THLE(1)
THLE(4) = THLE(3)

DTR = HT/1000.
DML = AMIN(H1,H2,H3)/1000.
MBITM = MBII-1
MBITP = MBII+1
MBOTM = MBOD-1
MBOTP = MBOD+1
MMIM = MMM-1
NRM = 0
CP = AR/(GAM-1.)*GAM
EXP = 1./(GAM-1.)
TWW = 2.*OMEGA/WTFL

CPTIP = 2.*CP*TIP
TGRUG = 2.*GAM*AR/(GAM+1.)

CALCULATE FINE MESH MV ARRAY

MV(1) = MV(MBIDF)
MBOT = MIN(MBIT,MBOD)
MBOT = MAX(MBOT,1)
DO 140 IM=1,MBOT
140 MV(IM) = MV(1)+FLOAT(IM-1)*HM1
MBIT = MAX(MBIT,MBOD)
MBIT = MIN(MBIT,MMM)
DO 150 IM=MBOT,MBIT
150 MV(IM) = MV(IM)+FLOAT(IM-MBOT)*HM2
DO 160 IM=MBIT,MMM
160 MV(IM) = MV(IM)+FLOAT(IM-MBIT)*HM3
DO 165 IM=1,MMM
165 IF (ABS(MV(IM)).LT.DML) MV(IM) = 0.
   CALL SPLINT(MR,MSP,NRSP,MV,MMM,RM,SAL)
   CALL SPLINT(MR,BESP,NRSP,MV,MMM,8,6,51)

FINISH READING NEW MAGNEY INPUT DATA
READ, COMPUTE, AND STORE BOUNDARY VALUES

DO 170 I=1,100
DO 170 J=1,4
   UBV(I,J) = 0.
170   RUBV(I,J) = 0.
180 READ(5,1010) KBDRY,NSP
IF (KBDRY.EQ.0) GO TO 210
WRITE(6,1320)
WRITE(6,1020) KBDRY,NSP
WRITE(6,1330)
READ (5,1030) (RVIN(I),I=1,NSP)
WRITE(6,1040) (BVIN(I),I=1,NSP)
WRITE(6,1340)
READ (5,1030) (UBVIN(I),I=1,NSP)
WRITE(6,1040) (UBVIN(I),I=1,NSP)
IF (KBDRY.EQ.3 .OR. KBDRY.EQ.4) GO TO 190
CALL BDVINT(BVIN,UBVIN,NSP,MV,KBDRY,DMLR,MMM)
GO TO 180
190 DO 200 IT=1,ITMAX
200 TH(IT)= FLOAT(IT-ITOR)*HT
CALL BDVINT(BVIN,UBVIN,NSP,TH,KBDRY,DTLR,ITMAX)
GO TO 180
210 DO 220 KBDRY=1,2
DO 220 ITM=1,MMM
220 RWBV(IM,KBDRY) = -RWBV(IM,KBDRY)*WTFL/BE(IM)
DO 230 KBDRY=3,4
DO 230 IT=1,ITMAX
230 RWBV(IT,KBDRY) = RWBV(IT,KBDRY)*WTFL/BE(IM)/RM(IM)
IF (HLDAT.GT.0) WRITE (6,1350) (MV(IM),UBV(IM),1,RWBV(IM,1),
1UBV(IM,2),RWBV(IM,2),IM=1,MMM)
IF (HLDAT.GT.0) WRITE (6,1360) (TH(IT),UBV(IT,3),RWBV(IT,3),
1UBV(IT,4),RWBV(IT,4),IT=1,ITMAX)
C
C INITIALIZE ARRAYS
C
DO 240 I=1,2000
U(I) = 1.
K(I) = 0.
240 RHO(I) = RHOIP
DO 250 IM=1,100
DO 250 SURF=1,4
RHOHB(IM,SURF) = RHOIP
RHUHB(IM,SURF) = RHOIP
RBV(IM,SURF) = RHOIP
250 ITV(IM,SURF) = -10000
DO 260 IM=1,100
ITV(IM,5) = 2
260 ITV(IM,6) = ITMAX-1
IF (MMM.LE.100.AND. ITMAX.LE.100) RETURN
WRITE (6,1370)
STOP
C
C FORMAT STATEMENTS
C
1000 FORMAT (1H1)
1010 FORMAT (16I5)
1020 FORMAT (1X,16I7)
1030 FORMAT (8F10.5)
1040 FORMAT (1X,8G16.7)
1100 FORMAT (80H
1)
1110 FORMAT (7X,4HNUBL/7X,13)
SUBROUTINE PRECAL
C
C PRECAL CALCULATES ALL REQUIRED FIXED CONSTANTS
C
1NOBL,MBI,MHU,MBI2,MB02,MM,NBBI,NBL,NRSP,MBDYF,MBDYL,ITF,ITL,
2BLDAT,ANUK,ERSDR,STRFV,INTVL,SURVL,MAGFAC,
3MR(50),RMSPI(50),BESP(50) COMMON /CALCON/ MBI,MBO1,MM1,MBI2,MHU1,MBI1,MHU1,MBI2,MHU2,
1MH,M2,MM,MN,DLR,DLMR,PITCH,CP,EXPON,TW,CPTIP,TRGOD,FB1,TB1,
2LAM:DA,ILW,ITOR,IATMX,INP,IM5(4),UV(4),MV(100),IV(100),
3SUB(100,4),RHBV(100,4),TV(100,4),TV(100,4),HUMV(100,4),
4BETV(100,4),MH(100,4),DDMH(100,4),BETAH(100,4),RMH(100,4),
5HEH(100,4),AM(100),BE(100),BDM(100),SAL(100),AAA(100)
INTEGER BLDAT,ANUK,ERSDR,STRFV,SURVL,AATEMP,SURF,
1FST,SUPP,SL,SRW
REAL K,KAK,LAMDA1,LMAX,MM,MLE,MR,MSP,MV,MMI1
EXTERNAL BLI,BL2,BL3,BL4
CALL CALCULATE TV, ITV, IV, DTDV AND BETAV ARRAYS

CALL TV, ITV, DTDV AND BETAV ON BLADES
SRW = 0
MBIT = MAXO(1,(MBI-MBDYF)*MAGFAC+1)
MBOT = MINO(MMM, MB00)
IF (MBOT.LT.MBIT) GO TO 20
DO 10 IM=MBIT,MBOT
LER(2) = 1
C BLCD CALL NO. 1
CALL BL1(MV(IM),TV(IM,1),DTDV(IM,1),INF)
    ITV(IM,1) = INT((TV(IM,1)+DTLR)/HT)+ITOR
    IF (TV(IM,1).GT.-DTLR) ITV(IM,1)=ITV(IM,1)+1
    ITV(IM,1) = MINO(ITV(IM,1),ITMAX)
    ITV(IM,1) = MAXO(ITV(IM,1),2)
    BETAV(IM,1)=ATAN(DTDV(IM,1)*RM(IM))*57.295779
    LER(2) = 2
C BLCD CALL NO. 2
CALL BL2(MV(IM),TV(IM,2),DTDV(IM,2),INF)
    ITV(IM,2) = INT((TV(IM,2)-DTLR)/HT)+ITOR
    IF (TV(IM,2).LT.DTLR) ITV(IM,2)=ITV(IM,2)-1
    ITV(IM,2) = MAXO(ITV(IM,2),1)
    ITV(IM,2) = MINO(ITV(IM,2),ITMAX-1)
10 BETAV(IM,2)=ATAN(DTDV(IM,2)*RM(IM))*57.295779
    MBIT = MAXO(1,MBII)
    MBOT = MINO(MMM,(MB32-MBDYF)*MAGFAC+1)
    IF (MBIT.GT.MBOT) GO TO 40
    DO 30 IM=MBIT,MBOT
    LER(2) = 3
C BLCD CALL NO. 3
CALL BL3(MV(IM),TV(IM,3),DTDV(IM,3),INF)
    ITV(IM,3) = INT((TV(IM,3)+DTLR)/HT)+ITOR
    IF (TV(IM,3).GT.-DTLR) ITV(IM,3)=ITV(IM,3)+1
    ITV(IM,3) = MINO(ITV(IM,3),ITMAX)
    ITV(IM,3) = MAXO(ITV(IM,3),2)
    BETAV(IM,3)=ATAN(DTDV(IM,3)*RM(IM))*57.295779
    LER(2) = 4
C BLCD CALL NO. 4
CALL BL4(MV(IM),TV(IM,4),DTDV(IM,4),INF)
    ITV(IM,4) = INT((Tv(IM,4)-DTLR)/HT)+ITOR
    IF (TV(IM,4).LT.DTLR) ITV(IM,4)=ITV(IM,4)-1
    ITV(IM,4) = MAXO(ITV(IM,4),1)
    ITV(IM,4) = MINO(ITV(IM,4),ITMAX-1)
30 BETAV(IM,4)=ATAN(DTDV(IM,4)*RM(IM))*57.295779
C IV ARRAY
40 IV(1) = 0
    IV(2) = 1
    MBOT = MINO(MBIIM1,MB00)
    IF (MBOT.LT.2) GO TO 60
    DO 50 IM=2,MBOT
    IV(I+1) = IV(IM)+ITV(IM,2)-ITV(IM,1)+ITMAX-1
50 IF (ITV(IM,1).EQ.-10000) IV(IM+1)=IV(IM+1)-1
    MBIT = MAXO(1,MBII)
    MBOT = MINO(MMM,MB00)
    IF (MBIT.GT.MBOT) GO TO 80
    DO 70 IM=MBIT,MBOT
    IV(IM+1) = TV(IM)+ITV(IM,4)+ITV(IM,2)-ITV(IM,3)-ITV(IM,1)+ITMAX
70 IF (IV(IM,1).EQ.-10000) IV(IM+1)=IV(IM+1)-1
    MBIT = MAXO(2,MB001)
    MBOT = MINO(MMM1,MBIIM1)
IF(MBIT.GT.MBOT) GO TO 100
DO 90 IM= MULIT,MBOT
90 IV(IM+1)= IV(IM)+ITMAX-2
100 MBIT = MAXO(MBI,MBOT)
IF(MBIT.GT.MMM1)GO TO 120
DO 110 IM=MBIT,MMM1
IV(IM+1)=IV(IM)*ITV(IM,4)-ITV(IM,3)+ITMAX-1
110 IF (ITV(IM,3).EQ.-10000) IV(IM+1)=IV(IM+1)-1
120 NIP= TV(MMM)-1
WRITE (6,1020) PITCH,HT,HM1,HM2,HM3
WRITE (6,1040) MBIT,MBOO,MMM,ITMAX,NIP
WRITE (6,1050) (SURF,DV(SURF),SURF=1,4)
IF(:LOAD.LF.0) GO TO 140
MBIT = MAXO(1,(MBI-MBDYF)*MAGFAC+1)
MBOT = MINO(MMM,MBOO)
WRITE (6,1050)
DO 130 SURF=1,3,2
I = SURF+1
IF (MBIT.LE.MBOT) WRITE(6,1060) SURF,I,(MV(IM),TV(IM,SURF),DTDMV(IM)
IM,SURF),TV(IM,I),DTDMV(IM,I),IM=MBIT,MBOT)
MBIT = MAXO(1,MBI)
130 MBOO = MINO(MMM,(MB02-MBDYF)*MAGFAC+1)
WRITE(6,1070) IM,MV(IM),TV(IM),MV(IM),PSL(IM),B3(IM),DBDM(IM),IM=1,MMM
WRITE(6,1080) IM,IV(IM),(ITV(IM,SURF),SURF=1,4),IM=1,MMM
C C CALCULATE MH AND DTMH ARRAYS.
C 140 IMS(1)=0
MRTS = 0
MBIT = MAXO(1,(MBI-MBDYF)*MAGFAC+1)
MBOT = MINO(MMM,MBOO)
LER(2) = 5
C BLC0 AND R0UT VIA WHORIZ) CALL NO. 5
CALL WHORIZ(MV,IV(1,1),HL1,MBIT,MBOT,ITOR,HT,UTLR,0,IMS(1),
IMH(1,1),DTDMH(1,1),MRTS)
IF(IV(MBOO,1)-ITV(MBOO,2).NE.2) GO TO 150
IMSL = IMS(1)+1
MH(MSL,1) = MV(MBOO)
DTDM(MSL,1) = -1.E10
IMS(1) = IMSL
150 IMS(2)=0
MRTS = 0
LER(2) = 6
C BLC0 AND R0UT VIA WHORIZ) CALL NO. 6
CALL WHORIZ(MV,IV(1,2),HL2,MBIT,MBOT,ITOK,HT,UTLR,1,IMS(2),
IMH(1,2),DTDMH(1,2),MRTS)
IMS(3)=0
IF(IV(MBI,3)-ITV(MBI,4).NE.2) GO TO 160
MRTS = 1
IMS(3)=1
MH(1,3) = MV(MBI)
DTDMH(1,3) = -1.E10
160 MBIT = MAXO(1,MBI)
MBOT = MINO(MMM,(MB02-MBDYF)*MAGFAC+1)
LER(2) = 7
C BLC0 AND R0UT VIA WHORIZ) CALL NO. 7
CALL WHORIZ(MV,IV(1,3),HL3,MBIT,MBOT,ITOK,HT,UTLR,0,IMS(3),
IMH(1,3),DTDMH(1,3),MRTS)
IMS(4)=0
IF (ITV(MBI,3)-ITV(MBI,4).EQ.2) MRTS=1
LER(2) = 8
BLCAND ROOT (VIA MHORIZ) CALL NO. 8
CALL MHORIZ(MV,ITV(1,4),HL4,MBIT,MBOT,ITOR,HT,DLR,1,IMS(4),
1MH(1,4),DTDMH(1,4),MRTS)
I = MAXO(IMS(1),IMS(2),IMS(3),IMS(4))
IF (I.LE.100) GO TO 170
WRITE(6,1090) I
STOP

C CALCULATE RMH, BHE, AND BETAH ARRAYS

170 IF (BLDATGT0) WRITE(6,1100)
DO 190 SURF=1,4
CALL SPLINT(MR,RMSP,NRSP,MH(1,SURF),IMS(SURF),RMH(1,SURF),AAA)
CALL SPLINT(MA,MBSP,MRSP,MH(1,SURF),IMS(SURF),DLH(1,SURF),AAA)
IMSS = IMS(SURF)
IF (IMSS.LT.1) GO TO 190
DO 180 IHS=1,IMSS
180 BETAH(IHSSURF) = ATAN(DTDMH(IHSSURF)*IMSS)
IF (BLDAT.LE.0) WRITE(6,1120)
DO 200 IT=1,ITMAX
200 WRITE(6,1200) IT,TH
210 WRITE(6,1100)
IF (NIP.LE.2000) RETURN
STOP

1000 FORMAT (1H1)
1010 FORMAT (4X,14,G16.5)
1020 FORMAT (1H1///5X,28HCALCULATED PROGRAM CONSTANTS//5X,5HPITCH,
113X,2HT,13X,3HHM1,13X,3HHM2,13X,3HHM3/1X,5G16.7)
1030 FORMAT (75X,4HM811,10X,4HMB00,10X,3HMM,10X,5HITMAX/3X,15,9X,15,
19X,15,9X,15///5X,33HNUMBER OF INTERIOR MESH POINTS = ,15)
1040 FORMAT (//////5X,23HSURFACE BOUNDARY VALUES//5X,7HSURFACE,7X,2HBV/
115X,4X,F10.5))
1050 FORMAT (1H1,6X,62HBLADE DATA AT INTERSECTIONS OF VERTICAL MESH LIN
1ES WITH BLADES)
1060 FORMAT (1H1///15X,13HBLADE SURFACE,I2,15X,13HBLADE SURFACE,I2/7X,
11HM,16X,2HTV,11X,5HDTDMV,12X,2HIV,11X,5HDTDMV/(5G15.5))
1070 FORMAT (1H1,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABL/
12X,2HIM,7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HDB/DMM/
2(1X,13,5G15.5))
1080 FORMAT (4H1 IM,9X,8HITV ARRAY/32X,9HITV ARRAY/38X,5HBLADE/37X,7HSUR
IFACE,3X,1H1,5X,1H2,5X,1H3,5X,1H4/39X,3HNO./1X,13,5X,110,25X,
24(414X)))
1090 FORMAT (35H1 ONE OF THE NH ARRAYS IS TOO LARGE/7HLLIT HAS,15,
18H POINTS)
1100 FORMAT (67HIM COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINE
15 WITH BLADE)
1110 FORMAT (25HLMH ARRAY - BLADE SURFACE,I2/15X,2HMH,19X,3HRMH,19X,
13HBEH,18X,5HBETAH,17X,5HDTDMV/(5G22.4))
1120 FORMAT (43HITHETA COORDINATES OF HORIZONTAL MESH LINES/6X,2HT,
15X,3HTHETA)
1130 FORMAT (48H THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2000)
END
SUBROUTINE COEF

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

COMMON SRW, ITER, IEND, LER(2), NER(1)
COMMON /INP/ GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, ORF, BETAI, BETAQ, INOB1, MB1, MBO, MB12, MBO2, MM, NBB1, NBL, NRS, MBDF, MBDY, ITF, ITL,
2BLDAT, AANDK, ERSOR, STRFN, INTVL, SURVL, MAGFAC,
3MR(50), RMSG(50), BESP(50)
COMMON /CALCON/ MB11, MBOOM, MMM, MBI1, MBI1P1, MBOOP1, MMM1,
1H11, HM2, HT1, DLTLR, DMLR, PITCH, CP, EXPON, TWW, CPTIP, TGROG, TBI, TBO,
2LAMBDA, TWL, ITUR, ITMAX, NIP, IMS(4), BVL(4), MV(100), IV(101),
3SUBV(100, 4), RMBV(100, 4), ITV(100, 4), TV(100, 4), DTDMV(100, 4),
4RETAV(100, 4), MH(100, 4), DTMH(100, 4), BETAH(100, 4), RMH(100, 4),
5BEH(100, 4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)
COMMON /HRBAAK/ H(4), R(4), B(4), KAK(4), KA(4), IH(4), RZ, BZ
INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1FIRST, UPPER, S1, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MR, MSP, MV, MVIM1

C INITIALIZE ARRAYS
ITER = ITER + 1
IH(1) = MAX0(0, ITV(2, 1) - ITV(1, 1))
IH(2) = MAX0(0, ITV(1, 2) - ITV(2, 2))
IH(3) = MAX0(0, ITV(2, 3) - ITV(1, 3))
IH(4) = MAX0(0, ITV(1, 4) - ITV(2, 4))
IF (ITV(MBI1, 3) - ITV(MBI1, 4) .EQ. 2) IH(3) = 1

C INCOMPRESSIBLE CASE
IF (GAM .NE. 1.5, OR, AR .NE. 1000, OR, TIP .NE. 1.6) GO TO 10
IEND = 0
GO TO 20

C ADJUSTMENT OF PRINTING CONTROL VARIABLES
10 IF (ITER .NE. 1, AND, ITER .NE. 2) GO TO 20
AANDK = AANDK - 1
ERSOR = ERSOR - 1
STRFN = STRFN - 1
INTVL = INTVL - 1
SURVL = SURVL - 1
20 IF (IEND .NE. 0) GO TO 30
AANDK = AANDK + 2
ERSOR = ERSOR + 2
STRFN = STRFN + 2
INTVL = INTVL + 2
SURVL = SURVL + 2

C CALL COEFBB THROUGHOUT THE REGION
C
C FRONT BLADE
30 MBOT = MIN0(MBI11, MBO0)
IF (MBOT .LT. 2) GO TO 50
DO 40 IM = 2, MBUT
CALL COEFBB (IM, 5, 2)
40 CALL COEFBB (IM, 1, 6)
C
C OVERLAP REGION
50 MBIT = MAX0(2, MBI1)
MBOT = MIN0(MMM1, MBO0)
IF (MBIT .GT. MBOT) GO TO 70
DO 60 IM = MBIT, MBOT
CALL COEFBB (IM, 5, 4)
60
CALL COEFBH (IM,3,2)
60 CALL COEFBH (IM,1,6)
GO TO 90
C NON-OVERLAP REGION
70 MBIT = MAXO(MII,MB00P1)
MBOT = MINO(MBIIM1,MMM1)
IF (MBIT.GT.MBOT) GO TO 90
DO 80 IM=MBIT,MBOT
80 CALL COEFBH (IM,5,6)
C REAR BLADE
90 MBIT = MAXO(MBI,MB00P1)
IF (MBIT.GT.MMMM1) GO TO 110
DO 100 IM=MBIT,MMMM1
CALL COEFBH (IM,5,4)
100 CALL COEFBH (IM,3,6)
C SPECIAL CASES - POINTS J OR C ARE MESH POINTS
C POINT J
110 IF (ITV(MBI,3)-ITV(MBI,4).NE.2) GO TO 120
IT = ITV(MBI,4)+1
IP = IPF(MBIIM1,IT)
K(IP) = K(IP)*A(IP,4)*BV(4)
A(IP,4) = 0.
C POINT C
120 IF(ITV(MBOO,1)-ITV(MBOO,2).NE.2) RETURN
IT = ITV(MBOO,2)+1
IP = IPF(MBOOP1,IT)
A(IP,3) = 0.
RETURN
END

SUBROUTINE COEFBB(IM,UPPER,LOWER)
C COEFBB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K
C ALONG ALL VERTICAL MESH LINES WHICH INTERSECT BLADES
C
COMMON /INP/ GAM,AR,TIP,RHOIP,WTL,WTFL,WTFLSP,OMEGA,ORF,BETA1,BETA0,
INOBL,MBI,MBO,MBI2,MB02,MM,MBII,NBL,NRSP,MBDYF,MBDYL,ITF,ITL,
2BLDAT,ANDK,ERSON,STRFN,INTVL,SURVL,MAGFAC,
3MR(50),RMSP(50),RESF(50)
COMMON /CALCON/ MBI,MBOO,MMM,MBIIM1,MBIIP1,MBOO1,MBOOP1,MMMM1,
1HM1,HM2,HM3,HT,DTLR,DLMR,PITCH,CP,EXPON,TW,CTP,PG1,MB01,
2LAMDA,TWL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RWBV(100,4),ITV(100,6),TV(100,4),DTV(100,4),
4BETAV(100,4),MH(100,4),DTMH(100,4),BETAH(100,4),RMH(100,4),
5BEHI(100,4),HM(100),BE(100),DBDM(100),SAL(100),AAA(100)
COMMON /HRHAAK/ H(4),R(4),B(4),KAK(4),KA(4),I(4),RZ,BZ
INTEGER BLDAT,ANDK,ERSON,STRFN,INTVL,SURVL,AATEMP,SURF,
1FIRST,UPPER,SL,ST,SRW
REAL K1,K1K,K1AMDA,LAMAX,MH,MLF,MG,MR,MSP,MV,MVIM1
IF(I(UPPER).GT.I(LOWER)) RETURN
ITVU= MAXO(I(UPPER),2)
ITVL= MINO(I(LOWER),ITMAX-1)
IF (ITVU.GT.ITVL) RETURN
IT = ITVU - 1
IPU = IPF(IM, ITVU)
IPL = IPU + ITVL - ITVU
DO 80 IP = IPU, IPL
IT = IT + 1
CALL HRB (IM, IT, IP)
DO 10 I = 1, 4
KAK(I) = 0.
10 KA(I) = 0
C
C FIX HRB VALUES FOR CASES WHERE MESH LINES INTERSECT BLADES OR BOUNDARIES
C
IF (IT .NE. 2) GO TO 20
KAK(1) = UBV(IM, 1)
KA(1) = 1
20 IF (IT .NE. ITMAX - 1) GO TO 30
KAK(2) = UBV(IM, 2)
KA(2) = 1
30 IF (IM .NE. 2) GO TO 40
KAK(3) = UBV(IT, 3)
KA(3) = 1
40 IF (IM .NE. MMAX1) GO TO 50
KAK(4) = UBV(IT, 4)
KA(4) = 1
50 IF (IT. EQ. ITVU .AND. UPPER .NE. 5) CALL BDRY12(1, IM, IT, UPPER)
IF (IT.EQ. ITVL .AND. LOWER .NE. 6) CALL BDRY12(2, IM, IT, LOWER)
ITV1 = ITV(IM - 1, UPPER)
ITVP1 = ITV(IM + 1, UPPER)
IF (ITV(IM, UPPER).EQ.-10000) GO TO 55
IF (IT.LT.ITVM1) CALL BDRY34(3, IM, UPPER)
IF (IT.LT.ITVP1) CALL BDRY34(4, IM, UPPER)
55 ITV1 = ITV(IM - 1, LOWER)
ITVP1 = ITV(IM + 1, LOWER)
IF (ITVM1.EQ.-10000) GO TO 60
IF (IM.EQ.MM1 .AND. LOWER .EQ. 4) GO TO 60
IF (IT.GT.ITVM1) CALL BDRY34(3, IM, LOWER)
60 IF (ITVP1.EQ.-10000) GO TO 70
IF (IM.EQ.MM0 .AND. LOWER .EQ. 2) GO TO 70
IF (IT.GT.ITVP1) CALL BDRY34(4, IM, LOWER)
C
C COMPUTE A AND K COEFFICIENTS
C
70 CALL AAK(IM, IP)
DO 80 I = 1, 4
K(IP) = K(IP) + KAK(I) * A(IP, I)
80 IF (KA(I).EQ.1) A(IP, I) = 0.
RETURN
END
SUBROUTINE NRB(IM, IT, IP)
C
C NRB CALCULATES MESH SPACING, H, DENSITIES, RZ AND R, AT GIVEN AND
C ADJACENT POINTS, AND STREAM SHEET THICKNESSES, BZ AND B, AT GIVEN
C AND ADJACENT POINTS
C
COMMON /CALCON/ MBII,MB00,MMM,MBIIIM,MBIIIP1,MBIO01,MBO001,MMM1,
IHMI,HM2,HM3,H1,DLTR,DLMR,PITCH,CP,EXPON,TW,CPTIP,TGROG,TBI,TBO,
2LAMDA,THL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RWBV(100,4),ITV(100,6),TV(100,4),DTMV(100,4),
4BETA(100,4),MH(100,4),DTHM(100,4),BETAH(100,4),RMH(100,4),
5BEH(100,4),RM(100),BE(100),DBDM(100),SA(100),AAA(100)
COMMON /RHOS/ RHOH5(100,4),RHOVB(100,4),RBV(100,4)
COMMON /HRAAAK/ H(1),R(4),B(4),KAK(4),KA(4),IH(4),RZ,BZ
INTEGER BLDAT,AANDK,ERSOR,STRFN,SURVL,AATEMP,SURF,
1FIRST,UPPER,SI,ST,SRW
REAL K,KAK,LAMDA,LMAX,MH,MLE,MR,MSP,MV,MVIM1
H(1)= HT*RM(IM)
H(2)= HT*RM(IM)
H(3)= MV(IM) - MV(IM-1)
H(4) = MV(IM+1) - MV(IM)
RZ = RHO(IP)
IP3 = IPF(IM-1,IT)
IP4 = IPF(IM+1,IT)
R(1) = RHO(IP-1)
IF (IT.EQ.2) R(1) = RBV(IM,1)
R(2) = RHO(IP+1)
IF (IT.EQ.ITMAX-1) R(2) = RBV(IM,2)
R(3) = RHO(IP3)
IF (IT.EQ.2) R(3) = RBV(IT,3)
R(4) = RHO(IP4)
IF (IM.EQ.MIM1) R(4) = RBV(IT,4)
BZ = BE(IM)
B(3) = BE(IM-1)
B(4) = BE(IM+1)
RETURN
FND

SUBROUTINE AAK(IM, IP)
C
C AAK CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANT, K,
C AT A SINGLE MESH POINT
C
COMMON /CALCON/ MBII,MB00,MMM,MBIIIM,MBIIIP1,MBIO01,MBO001,MMM1,
IHMI,HM2,HM3,H1,DLTR,DLMR,PITCH,CP,EXPON,TW,CPTIP,TGROG,TBI,TBO,
2LAMDA,THL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RWBV(100,4),ITV(100,6),TV(100,4),DTMV(100,4),
4BETA(100,4),MH(100,4),DTHM(100,4),BETAH(100,4),RMH(100,4),
5BEH(100,4),RM(100),BE(100),DBDM(100),SA(100),AAA(100)
COMMON /RHOS/ RHOH5(100,4),RHOVB(100,4),RBV(100,4)
COMMON /HRAAAK/ H(1),R(4),B(4),KAK(4),KA(4),IH(4),RZ,BZ
INTEGER BLDAT,AANDK,ERSOR,STRFN,SURVL,AATEMP,SURF,
1FIRST,UPPER,SI,ST,SRW
REAL K,KAK,LAMDA,LMAX,MH,MLE,MR,MSP,MV,MVIM1
SUBROUTINE GORY12 (I, IM, IT, SURF)

C C GORY12 CORRECTS VALUES COMPUTED BY HRB WHEN A VERTICAL MESH LINE
C INTERSECTS A BLADE
C
C COMMON /CALCIN/ MBII, MB00, MMM, MBII, MBII, MBII, MB00, MMM, MMM,
1 HM1, HM2, HM3, HT, DTLR, DMLR, PITCH, CP, EXPON, TWW, CTP, CP, TROG, TBI, TBO,
2 LAMDA, TWL, ITOR, ITMAX, NIP, IMS, BV, MV, IV(100), IV(101),
3 BV(100), IV(100), TV(100), TV(100), TV(100), TV(100), TV(100), TV(100),
4 BETA(100), MH(100), MH(100), STR(100), TV(100), TV(100),
5 BETA(100), MH(100), MH(100), BMV, MV, MV,
C COMMON /RHOS/ RHOS(100, 4), RHOS(100, 4), RHOS(100, 4), RHOS(100, 4),
COMMON /HRBAAK/ H(4), R(4), B(4), M(4), R(4), M(4), R(4), M(4),
INTEGER BLDT, AOND, ER, STRF, SURV, AATR, SURF,
1 FIRST, UPPER, SI, ST, SR
REAL K, KAK, LAMDA, LMAX, MH, MLE, MR, MSP, MV, MVIM
H(I) = ABS(FLOAT(IT-ITOR)*HT-TV(IM, SURF)) * RM(1M)
K(I) = RV(SURF)
KAK(I) = BV(SURF)
KAK(I) = 1
RETURN
END

SUBROUTINE GORY34(I, IM, SURF)

C C GORY34 CORRECTS VALUES COMPUTED BY HRB WHEN A HORIZONTAL MESH LINE
C INTERSECTS A BLADE
C
C COMMON /CALCIN/ MBII, MB00, MMM, MBII, MBII, MBII, MB00, MMM, MMM,
1 HM1, HM2, HM3, HT, DTLR, DMLR, PITCH, CP, EXPON, TWW, CTP, CP, TROG, TBI, TBO,
2 LAMDA, TWL, ITOR, ITMAX, NIP, IMS, BV, MV, IV(100), IV(101),
3 BV(100), IV(100), TV(100), TV(100), TV(100), TV(100), TV(100), TV(100),
4 BETA(100), MH(100), MH(100), STR(100), TV(100), TV(100),
5 BETA(100), MH(100), MH(100), BMV, MV, MV,
C COMMON /RHOS/ RHOS(100, 4), RHOS(100, 4), RHOS(100, 4), RHOS(100, 4),
COMMON /HRBAAK/ H(4), R(4), B(4), M(4), R(4), M(4), R(4), M(4),
INTEGER BLDT, AOND, ER, STRF, SURV, AATR, SURF,
1 FIRST, UPPER, SI, ST, SR
REAL K, KAK, LAMDA, LMAX, MH, MLE, MR, MSP, MV, MVIM
IH(SURF) = IH(SURF) + 1
IH(SURF)
H(I) = ABS(MV(IM) - MH(IHS, SURF))
R(I) = RHOHB(IHS, SURF)
B(I) = BEH(IHS, SURF)
KAK(I) = BV(SURF)
KA(I) = 1
RETURN
END

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

COMMON /INP/ GAM, AR, TIP, RHOIIP, WTI, WTI, OMEGA, OR, BETAI, BETAO,
1NOB, MBI, MB0, MBI2, MB02, MIBI, NBL, NRSP, MBDYF, MBDSL, ITF, ITL,
2BDAT, AANDK, ERSOR, STRFN, INTVL, SURVL, MAGFAC,
3MR(50), RSP(50), BESP(50)
COMMON /CALCON/ MBI, MB00, MMM, MBIM1, MB1P1, MB00M1, MB0O1, MMM1,
1H1, HM2, HM3, HI, DTLR, DMLR, PITCH, CP, EXPON, TWW, CPTIP, TGRUG, TB1, TBO,
2LAMDA, TWL, ITD, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3UBV(100, 4), RKBV(100, 4), ITV(100, 6), TV(100, 4), DTDMV(100, 4),
4BETA(100, 4), MH(100, 4), DTDMH(100, 4), BETAH(100, 4), RM(100, 4),
5BEH(100, 4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1FIRST, UPPER, SL, ST, SRW
REAL K, KAK, LAMDA, LMAX, MH, MLE, MR, MSP, MV, MVIM1
AATEMP = AANDK
IF (ORF.GE.2) ORF = 0.
IF(ORF.GT.1) GO TO 20
ORF = 1.
ORFOPT = 2.
10 ORFTEM = ORFOPT
LMAX = 0.
20 IF(AATEMP.GT.0) WRITE(6,1010)
ERROR = 0.

SOLVE MATRIX EQUATION BY SOR, OR CALCULATE OPTIMUM OVERRELAXATION FACTOR

DO 40 IM=2, MMM1
IF(AATEMP.GT.0) WRITE(6, 1020) IM
IPU = IV(IM)
IPL = IV(IM+1) - 1
IT = 1
DO 50 IP=IPU, IPL
IF(IPU.GT.IPL) GO TO 50
IT = IT + 1
IF(IT.GT.ITV(IM, 4).AND.IT.LE.ITV(IM, 3)) IT = IT + ITV(IM, 3) -
1MAXO(ITV(IM, 4), 1) - 1
IF(IT.GT.ITV(IM, 2).AND.IT.LE.ITV(IM, 1)) IT = IT + ITV(IM, 1) -
1MAXO(ITV(IM, 2), 1) - 1

57
IP1 = IP-1
IP2 = IP+1
IP3 = IPF(IM-1, IT)
IP4 = IPF(IM+1, IT)
IF(IM.EQ.2) IP3 = 0
IF(IM.EQ.MMIM1) IP4 = 0
IF(ORF.GT.1.) GO TO 30
C CALCULATE NEW ESTIMATE FOR LMAX
UNEW = A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4)
IF (UNEW.LE.1.E-25) U(IP) = 0.
IF (U(IP).EQ.0.) GO TO 40
RATIO = UNEW/U(IP)
LMAX = AMAX1(RATIO, LMAX)
U(IP) = UNEW
GO TO 40
C CALCULATE NEW ESTIMATE FOR STREAM FUNCTION BY SOR
30 CHANGE = ORF*(K(IP)-U(IP)+A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*
1U(IP3)+A(IP,4)*U(IP4))
ERROR= AMAX1(ERROR, ABS(CHANGE))
U(IP) = U(IP)+CHANGE
40 IF(AATEMP.LE.0) GO TO 50
WRITE (6,1030) IT, IP, IP1, IP2, IP3, IP4, (A(IP,I), I=1,4), K(IP)
50 CONTINUE
AATEMP = 0
IF(ORF.GT.1.) GO TO 60
ORFOPT = 2./(1.+SQRT(AAS(1.-LMAX)))
WRITE (6,1040) ORFOPT
IF (ORFTEM-URFOPT.GT.00001.OR.URFOPT.GT.1.999) GO TO 10
WRITE (6,1000)
ORF = ORFOPT
GO TO 20
60 IF(ERROR.GT.0) WRITE(6,1050) ERROR
IF(ERROR.GT.000001) GO TO 20
IF(STRFN.LE.0) RETURN
C
C PRINT STREAM FUNCTION VALUES FOR THIS ITERATION
C
WRITE (6,1060)
IPL = 0
MBOT = MINO(MBIIM1, MBCO)
IF (MBOT.LT.2) GO TO 80
DO 70 IM=2, MBOT
CALL WRITU(IM,5,2, IPL)
70 CALL WRITU(IM,1,6, IPL)
80 MBIT = MAXO(2, MBII)
MBOT = MINO(MMMIM1, MBOO)
IF (MBIT.GT.MBOT) GO TO 100
DO 90 IM=MBIT, MBOT
CALL WRITU(IM,5,4, IPL)
CALL WRITU(IM,3,2, IPL)
90 CALL WRITU(IM,1,6, IPL)
GO TO 120
100 MBIT = MAXO(2, MBOOP1)
MBOT = MINO(MBIIM1, MMMIM1)
IF (MBIT.GT.MBOT) GO TO 120
DO 110 IM=MBIT, MBOT
110 CALL WRITU(IM,5,6, IPL)
120 MBIT = MAXO(MBIIM, MBOOP1)
IF (MBIT.GT.MMMM1) RETURN
DO 130 IM=MBIT,MMMM1
CALL WRITU(IM,5,4,IPL)
130 CALL WRITU(IM,3,6,IPL)
RETURN
1000 FORMAT (1H1)
1010 FORMAT (82HIT IP IPL IP2 IP3 IP4 A(1) A(2))  
1 A(3) A(4) K)
1020 FORMAT(5H IM =,14)
1030 FORMAT(1X,14,5I6,5F10.5)
1040 FORMAT(24H ESTIMATED OPTIMUM ORG =,F9.6)
1050 FORMAT(8H ERROR =,F11.8)
1060 FORMAT(1HlT10X122HSTREAM FUNCTION VALUES)
END

SUBROUTINE WRITU(IM,UPPER,LOWER,IPL)
COMMON /CALCON/ MBII,MB00,MMM,MBIIPL,MB00PL,MMM,MBII,MB00,MMM,MBIIPL,MB00PL,MMM,
1HM1,HM2,HM3,HT,DTLR,DMLR,PITCH,CP,EXPON,TWN,CPTIP,TRGLD,TBI,TBO,
2LAMBDATWL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RNBV(100,4),ITV(100,4),TV(100,4),DTDMV(100,4),
4BETA(100,4),MH(100,4),DTDMH(100,4),BETAH(100,4),RMH(100,4),
5BEH(100,4),RM(100),BE(100),DBDM(100),SAL(100),AAA(100)
INTEGER BLDAT,AANDK,ERSOR,STRFN,SURVL,AATEMP,SURF,
1FIRST,UPPER,SL,ST,SRW
ITVU = MAX0(ITV(IM,UPPER),2)
ITVL = MIN0(ITV(IM,LOWER),ITMAX-1)
IF(ITVU.GT.ITVL) RETURN
IPU = IPL+I
IPL = IPU+ITVL-ITVU
WRITE(6,1000) IM,ITVU
WRITE(6,1010) (U(IP),IP=IPU,IPL)
RETURN
1000 FORMAT(5H IM =,13,10X,5HIT1 =,13)
1010 FORMAT (2X,10F13.8)
END

SUBROUTINE SLAX
C
C SLAX CALLS SUBROUTINES TO CALCULATE RHO*W-SUB-M THROUGHOUT THE REGION
C AND ON THE BLADE SURFACES, AND TO CALCULATE AND PLOT THE
C STREAMLINE LOCATIONS
C
COMMON /INP/ GAM,AR,TIP,RHOIP,WTFL,WNFLSP,OMEGA,ORF,BETA,BETAO,
1NBL,MBI,MBO,MBI2,MBO2,MMM,NBBI,NBL,NRSF,MBDYF,MBDYL,ITF,ITL,
2BLDAT,AANDK,ERSOR,STRFN,INTVL,SURVL,MAGFAC,
3MR(50),RMSP(50),BESP(50)
COMMON /CALCON/ MBII,MB00,MMM,MBIIPL,MB00PL,MMM,MBII,MB00,MMM,MBIIPL,MB00PL,MMM,
1HM1,HM2,HM3,HT,DTLR,DMLR,PITCH,CP,EXPON,TWN,CPTIP,TRGLD,TBI,TBO,
2LAMBDATWL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RNBV(100,4),ITV(100,4),TV(100,4),DTDMV(100,4),
DO 10 I=1,4
DO 10 IM=1,100
  WMB(IM,I) = 0.
10 WTB(IM,I) = 0.
  MBOT = MINO(MBI11M1,MB00)
  IF (MBOT.LT.2) GO TO 30
  DO 20 IM=2,MBOT
     CALL SLAVBB(IM,5,2)
20 CALL SLAVBR(IM,1,6)
  MBIT = MAXO(2,MBII)
  MBOT = MINO(MMM111,MB00)
  IF (MBIT.GT.MBOT) GO TO 50
  DO 40 IM=MBIT,MBOT
     CALL SLAVBB(IM,5,4)
     CALL SLAVBR(IM,3,2)
40 CALL SLAVBB(IM,1,6)
  GO TO 70
  MBIT = MAXO(2,MB00P1)
  MBOT = MINO(MBI11M1,MMM111)
  IF (MBIT.GT.MBOT) GO TO 70
  DO 60 IM=MBIT,MBOT
60 CALL SLAVBB(IM,5,6)
  MBIT = MAXO(MBI11,MB00P1)
  IF (MBIT.GT.MMM111) RETURN
  DO 80 IM=MBIT,MMM111
80 CALL SLAVBR(IM,5,4)
  RETURN
END
SUBROUTINE SLAVBB(ILM, UPPER, LOWER)

SLAVBB calculates RHO*W-SUB-M along vertical mesh lines.

COMMON SRW, ITER, IEND, LER(2), NER(1)
COMMON /INP/ GAM, AR, TIP, RHOIP, WFL, TFL, LSP, OMEGA, ORF, BETAI, BETA0,
1NOBL, MB1, MB2, MB3, MM, NBBI, NBL, NRSP, MBDFI, MBDYL, ITF, ITL,
2BLDAT, AANDK, ERSOR, STRFN, INTVL, SURVL, MAGFAC,
MR(50), RMSP(50), BESP(50)
COMMON /CALC/ MBII, MBGO, MMM, MBII1M, MBII1I, MBGO1M, MBGO1I, MM1M1,
1HMI, HM2, HM3, HT, DTTR, DMLR, PITCH, CP, EXPON, TWH, CPTIP, TGROG, TBI, TBO,
2LAMBDA, TWL, ITOR, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3UBV(100,4), RWBV(100,4), TV(100,6), TVL(100,4), DTMV(100,4),
4BETA(100,4), MHI(100,4), DTMH(100,4), BETAH(100,4), RMH(100,4),
5BHH(100,4), RM(100), BE(100), DBDM(100), SA1(100), AAA(100)
COMMON /RHS/ RHOHB(100,4), RHOVB(100,4), RBV(100,4),
DIMENSION W(2000), RWM(2000), ETA(2000), WMB(100,4), WTB(100,4),
1XDOWN(800), YACROS(800), TSL(400), TSP(100), USP(100), DUDT(100)
EQUIVALENCE (A(1,1), W(1)), (A(1,2), RWM(1)), (A(1,3), ETA(1)),
1(A(1,4), WMB(1)), (A(401,4), WTB(1)), (A(801,4), XDOWN(1)),
2(K(1), YACROS(1)), (K(801), TSL(1)),
3(K(901), USP(1)), (K(1001), DUDT(1))
INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1FIRST, UPPER, SL, ST, SRW
REAL K, KAK, LAMBDA, LMAX, MH, MLE, MK, MSP, MV, MM1M
ITVU = MAX0(ITVI(1, UPPER), 2)
ITVL = MINO(ITVI(1, LOWER), ITMAX-1)
NSP = ITVL-ITVU+3
IF(VSP.LT.3) RETURN
TSP(1) = FLOAT(1-ITOR) * HT
IF(ITVI(1, UPPER).LT.2 OR UPPER.EQ.5) GO TO 10
IF(TVI(1, UPPER).LT.1) GO TO 10
TSP(1) = TVI(1, UPPER)
USP(1) = BV(UPPER)
GO TO 20
10 USP(1) = UBV(1, 1)
20 TSP(NSP) = FLOAT(ITMAX-ITOR) * HT
IF(ITVI(1, LOWER).GE.ITMAX OR LOWER.EQ.6) GO TO 30
IF (TVI(1, LOWER).GE.TSP(NSP)) GO TO 30
TSP(NSP) = TVI(1, LOWER)
USP(NSP) = RV(LOWER)
GO TO 40
30 USP(NSP) = UBV(1, 2)
40 NSPM1 = NSP-1
IT = 2
IP = IFP(1, ITVU)
IPU = IP
50 IF(IT.GT.NSPM1) GO TO 60
TSP(IT) = FLOAT(IT-2+ITVU-ITOR) * HT
USP(IT) = U(IP)
IT = IT+1
IP = IP+1
GO TO 50

CALL SPLINE(TSP, USP, NSP, DUDT, AAA)
IPL = IP - 1
IT = 2
IP = IPU
70 IF (IP .GT. IPL) GO TO 80
    RWMP(IP) = DUDT(IT) * WTFL/BE(IM)/RM(IM)
    IP = IP + 1
    IT = IT + 1
    GO TO 70
80 IF (ITV(IM, UPPER) .LT. 2. OR. UPPER .EQ. 5) GO TO 90
    IF (T(V(IM, UPPER) .LT. FLOAT(1 - ITOR) * HT) GO TO 90

    : UPPER BLADE SURFACE
    WMB(IM, UPPER) = DUDT(1) * WTFL/BE(IM)/RM(IM)
    RMDTU2 = (RM(IM) * DTDMV(IM, UPPER))**2
    IF (RMDTU2 .LT. 10000) WMB(IM, UPPER) = 0.
    WMB(IM, UPPER) = ABS(WMB(IM, UPPER)) * SQRT(1. + RMDTU2)
    GO TO 100
C LOWER BOUNDARY
90 RWMBV = DUDT(1) * WTFL/BE(IM)/RM(IM)
    RW = SQRT(RWMBV(IM, 1)**2 + RWMBV**2)
    TWLMR = 2. * OMEGA * LAMBDA - (OMEGA * RM(IM))**2
    LER(1) = 1
C DENSITY CALL NO. 1
    CALL DENSITY(RW, RBV(IM, 1), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
100 IF (ITV(IM, LOWER) .GE. ITMAX OR LOWER .EQ. 6) GO TO 110
    IF (T(V(IM, LOWER) .GE. FLOAT(1 - ITOR) * HT) GO TO 110

    : LOWER BLADE SURFACE
    WMB(IM, LOWER) = DUDT(NSP) * WTFL/BE(IM)/RM(IM)
    RMDTL2 = (RM(IM) * DTDMV(IM, LOWER))**2
    IF (RMDTL2 .LT. 10000) WMB(IM, LOWER) = 0.
    WMB(IM, LOWER) = ABS(WMB(IM, LOWER)) * SQRT(1. + RMDTL2)
    RETURN
C UPPER BOUNDARY
110 RWMBV = DUDT(NSP) * WTFL/BE(IM)/RM(IM)
    RW = SQRT(RWMBV(IM, 2)**2 + RWMBV**2)
    TWLMR = 2. * OMEGA * LAMBDA - (OMEGA * RM(IM))**2
    LER(1) = 2
C DENSITY CALL NO. 2
    CALL DENSITY(RW, RBV(IM, 2), ANS, TWLMR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
    RETURN
END

SUBROUTINE TANG

C TANG CALCULATES RHO*W-SUB-THETA AND THEN RHO*W THROUGHOUT THE REGION
C AND ON THE BLADE SURFACES, AND CALCULATES THE VELOCITY ANGLE, BETA,
C THROUGHOUT THE REGION
C
COMMON SRW, ITLR, LND, LER(2), VER(1)
COMMON /IMP/ GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, OKF, BELTAI, BETAU,
1NOBL, MBI, MCI, MBI2, MBO2, MKM, NBI, NRNL, NRSP, MBDYF, MBDYL, ITF, ITL,
2BLDATE, ANUAK, EKSRU, STREN, INTVL, SURVL, MAGFA, 3MK(50), RMS(50), BESP(50)
COMMON /CALC/ MBBI, MBIG, MM, MBIM1, MBIP, MBOP1, MM1,  
1HM1, HM2, HM3, HT, DLR, MR, PITCH, CP, EXPON, TPH, CPTIP, TETRO, TBU, TBL,  
2LAMBDA, TWL, ATE, ITM, NIP, INS1, BV1, MV1, IV(101),  
3UBV(100,4), RUBV(100,4), ITV(100,4), TV(100,4), DTDMV(100,4),  
4TBAV(100,4), MS(100,4), DTDM(100,4), TBAH(100,4), RM(100,4),  
5MHE(100,4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)  
COMMON /RHO/ RHOB(100,4), RADOB(100,4), RBV(100,4)  
DIMENSION SM(100), USP(100), DU(100)  
DIMENSION W(2000), RWM(100,4), BETA(1000), WMB(100,4), WTB(100,4),  
1XDOWN(800), YACROS(800)  
EQUIVALENCE (A(1,1),W(1)), (A(1,2),RWM(1)), (A(1,3),BETA(1)),  
1(A(1,4),WMB(1)), (A(401,4), WTB(1)), (A(1001,4), XDOWN(1)),  
2(K(1), YACROS(1))  
INTEGER BL1, BL2, BL3, BL4  
REAL K, FBL, LAMBDA, LMAX, MH, MLE, MR, MSP, MV, MVIM1  
EXTERNAL BL1, BL2, BL3, BL4  
  
P= 2  
10 IF (IT.EQ.IMAX) RETURN  
  
C GIVEN HORIZONTAL MESH LINE, FIND FIRST POINT IN THE REGION  
C  
IF (MBI1.LE.1 .AND. IT.LE.ITV(1,1) .OR. IT.GE.ITV(1,3) .AND.  
1IT.LE.ITV(1,2)) .OR. IT.GE.ITV(1,3)) GO TO 50  
IF (MBII.GT.2 .AND. MBII.LT.0 .OR. IT.LE.ITV(1,2) .OR.  
1IT.GE.ITV(1,1)) GO TO 50  
IM = 1  
20 IM = IM + 1  
IF (IM.GE.MM) GO TO 180  
DO 30 SURF = 1, 2  
IF (IM.GT.MBO .AND. SURF.EQ.1) GO TO 30  
IF (IM.LE.MBI .AND. SURF.EQ.3) GO TO 30  
IF (IT.GE.ITV(IM,SURF).AND.IT.LT.ITV(IM-1,SURF)) GO TO 60  
30 CONTINUE  
SURF = 1  
IF (IM.EQ.MBO .AND. IT.GE.ITV(MBO,1)-1 .AND.  
1ITV(MBO,1)-ITV(MBO,2) .EQ. 2)) GO TO 60  
DO 40 SURF = 3, 4, 2  
IF (IM.GT.MBO .AND. SURF.EQ.2) GO TO 40  
IF (IM.LE.MBI .AND. SURF.EQ.4) GO TO 40  
IF (IT.GE.ITV(IM,SURF).AND.IT.GT.ITV(IM-1,SURF)) GO TO 60  
40 CONTINUE  
GO TO 20  
  
FIRST POINT IS ON LEFT BOUNDARY  
  
50 S1 = 0  
IM = 1  
IM = 2  
SPM(1) = MV(1)  
USP(1) = UBV(IT,3)  
GO TO 70  
  
FIRST POINT IS ON A BLADE SURFACE  
  
60 S1 = SURF  

IM1 = IM - 1  
IM2 = IM  
TH = FLOAT(IT - ITO) * HT  
MVIM1 = MV(IM1)  
IF (IM.EQ.IM1.AND.(SURF.EQ.3.OR.SURF.EQ.4)) MVIM1 = MV(IM2) - MVIM1 / 1000 .  
LER(2) = 9  
C BLCD (VIA Rcoding) CALL NO. 9  
IF (S1.EQ.1.AND.IM1.NE.MM00) CALL ROOT(MVIM1, MV(IM2), TH, BL1,  
1DTLR, ANS, AAA)  
LER(2) = 10  
C BLCD (VIA Rcoding) CALL NO. 10  
IF (S1.EQ.2) CALL ROOT(MVIM1, MV(IM2), TH, BL2, DTLR, ANS, AAA)  
LER(2) = 11  
C BLCD (VIA Rcoding) CALL NO. 11  
IF (S1.EQ.3) CALL ROOT(MVIM1, MV(IM2), TH, BL3, DTLR, ANS, AAA)  
LER(2) = 12  
C BLCD (VIA ROOT) CALL NO. 12  
IF (S1.EQ.4) CALL ROOT(MVIM1, MV(IM2), TH, BL4, DTLR, ANS, AAA)  
IF (S1.EQ.1.AND.IM1.EQ.MM00) ANS = MV(MM00)  
SPM(IM1) = ANS  
USP(IM1) = IP(S1)  
C  
MOV P ALONG HORIZONTAL MESH LINE UNTIL END OF REGION IS REACHED  
C  
70 DO DO SURF = 1, 3, 2  
IF (IM.GT.MM00.AND.SURF.EQ.1) GO TO 80  
IF (IM.LE.MM1.AND.SURF.EQ.3) GO TO 80  
IF (ITV(IM-1, SURF).EQ.-10000) GO TO 80  
IF (IT.LT.I TV(IM, SURF).AND.IT.GE.I TV(IM-1, SURF)) GO TO 110  
80 CONTINUE  
SURF = 3  
IF (IM.EQ.MM1.AND.IT.EQ.I TV(MM1, 3)-1.AND.  
ITV(MM1, 3)-ITV(MM1, 4).EQ.2) GO TO 110  
DO 90 SURF = 2, 4, 2  
IF (IM.GT.MM00.AND.SURF.EQ.2) GO TO 90  
IF (IM.LE.MM1.AND.SURF.EQ.4) GO TO 90  
IF (IT.GT.I TV(IM, SURF).AND.IT.LT.I TV(IM-1, SURF)) GO TO 110  
90 CONTINUE  
SPM(IM) = MV(IM)  
IP = IPF(IM, IT)  
USP(IM) = U(IP)  
IF (IM.EQ.MM00) GO TO 100  
IM = IM + 1  
GO TO 70  
C  
FINAL POINT IS ON RIGHT BOUNDARY  
C  
100 ST = 0  
IMT = MMM  
USP(IMT) = UBV(IT, 4)  
GO TO 120  
C  
FINAL POINT IS ON A BLADE SURFACE  
C  
110 ST= SURF  
IMT = IM  
IMTM1 = IMT - 1  
TH = FLOAT(IT - ITO) * HT  
MVIM1 = MV(IMTM1)
IF (IMT1 .EQ. MBII) .AND. (ST .EQ. 3 .OR. ST .EQ. 4) .AND. (ITV(MBII, 3) - 1 .EQ. (ITV(MBII, 4) .EQ. 2)) MVIM1 = MVIM1 + (MV(IMT) - MVIM1) / 1000.

LER(2) = 13

C BLCC (VIA ROOT) CALL NO. 13
IF (ST .EQ. 1) CALL ROOT(MVIM1, MV(IMT), TH, BL1, DTLR, ANS, AAA)
LER(2) = 14

C BLCC (VIA ROOT) CALL NO. 14
IF (ST .EQ. 2) CALL ROOT(MVIM1, MV(IMT), TH, BL2, DTLR, ANS, AAA)
LER(2) = 15

C BLCC (VIA ROOT) CALL NO. 15
IF (ST .EQ. 3 .AND. IMT .NE. MBII) CALL ROOT(MVIM1, MV(IMT), TH, BL3, 10TLR, ANS, AAA)
LER(2) = 16

C BLCC (VIA ROOT) CALL NO. 16
IF (ST .EQ. 4) CALL ROOT(MVIM1, MV(IMT), TH, BL4, DTLR, ANS, AAA)
IF (ST .EQ. 3 .AND. IMT .EQ. MBII) ANS = MV(MBII)
SPM(IMT) = ANS
USP(IMT) = 8V(ST)

C CALCULATE RHC*W-SUB-THETA AND THEN RHC*W AND BETA IN THE REGION

C 120 NSP = IMT - IML + 1
CALL SPLINE(SPM(IMT), USP(IMT), NSP, DUDM(IMT), AAA(IMT))
FIRST = 2
IF (IMT .NE. 1) FIRST = IM2
LAST = MMMM1
IF (IMT .NE. KMM) LAST = IMTM1
IF (FIRST .EQ. LAST) GO TO 140
DO 130 I = FIRST, LAST
RWT = -DUDM(I) * WTFL / RE(I)
IP = IPF(I, IT)
W(IP) = SQRT(RWT**2 + RWB(IP)**2)
130 BET(IP) = ATAN(RWT / RWB(IP)) * 57.295779

C CALCULATE RHO*W ON THE BLADE SURFACES, OR RHO ON VERTICAL BOUNDARIES

C 140 IF (ST .EQ. 0) GO TO 150
CALL SEARCH(SPM(IMT), S1, IHS)
ANS = -DUDM(IMT1) * WTFL / REH(IHS, S1)
WTB(IHS, S1) = ASANS * SQRT(1. + 1. / (RMH(IHS, S1) * DTDMH(IHS, S1))**2)
GO TO 160
150 RWT = -DUDM(I) * WTFL / BE(I)
RWT = SQRT(RWT**2 + RWB(V(1T), 3)**2)
TWLMR = 2 * CMega * LAMDA - (CMega * RM(1))**2
LER(1) = 3

C DENSITY CALL NO. 3
CALL DENSTY(RW, REV(IT, 3), ANS, TWR, CPTIP, EXPON, RHOIP, GAM, AR, TIP)
160 IF (ST .EQ. 0) GO TO 170
CALL SEARCH(SPM(IMT), ST, IHS)
ANS = -DUDM(IMT) * WTFL / BEH(IHS, ST)
WTB(IHS, ST) = AS Anita * SQRT(1. + 1. / (RMH(IHS, ST) * DTDMH(IHS, ST))**2)
GO TO 20
170 RWT = -DUDM(MMM) * WTFL / BE(MMM)
RWT = SQRT(RWT**2 + RWB(V(1T, 4)**2)
TWLMR = 2 * CMega * LAMDA - (CMega * RM(MMM))**2
LER(1) = 4

C DENSITY CALL NO. 4
CALL DENSITY(RW, R8V(IT, 4), ANS, TWR, CPTIP, EXPUN, RHOIP, GAM, AR, TIP)
180 IT = IT + 1
GO TO 10
END
SUBROUTINE SEARCH (DIST, SURF, IS)

SEARCH LOCATES THE POSITION OF A GIVEN VALUE OF M IN THE MH ARRAY

COMMON /CALCON/ MBII, MBOO, MMM, MBIIM1, MBIIP1, MBOOM1, MBOOP1, MMMM1,
1HM1, HM2, HM3, HT, DTLR, DMLR, PITCH, CP, EXPON, TW, CPTIP, TGROG, TBI, TBO,
2LAMODA, TWL, ITOR, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3UBV(100,4), RWBV(100,4), ITV(100,6), TV(100,4), DTMV(100,4),
4BETA(100,4), MH(100,4), DTMH(100,4), BETAH(100,4), RMH(100,4),
5BEH(100,4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)

INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1FIRST, UPPER, S1, ST, SRW

REAL K, KAK, LAMBOA, LMAX, MH, MLE, MR, MSP, MV, MVIM1

DO 10 I=1, IV(101)

IF (ABS(MH(I,SURF) - DIST) .GT. DMLR) GO TO 10

IS = I

RETURN

10 CONTINUE

WRITE (6, 1000) DIST, SURF

STOP

1000 FORMAT (38HL SEARCH CANNOT FIND M IN THE MH ARRAY/7H DIST =, G14.6,
110X, 6HSURF =, G14.6)

END

SUBROUTINE VELOCY

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


COMMON /INP/ GAM, AR, TIP, RHOIP, WFL, WFLSP, OMEGA, ORF, BETAI, BETAO,
INOB, MBII, MBII, MBII, MBII, MBII, MBII, MBII, MBII, MBII, MBII, MBII,
1HM1, HM2, HM3, HT, DTLR, DMLR, PITCH, CP, EXPON, TW, CPTIP, TGROG, TBI, TBO,
2LAMODA, TWL, ITOR, ITMAX, NIP, IMS(4), BV(4), MV(100), IV(101),
3UBV(100,4), RWBV(100,4), ITV(100,6), TV(100,4), DTMV(100,4),
4BETA(100,4), MH(100,4), DTMH(100,4), BETAH(100,4), RMH(100,4),
5BEH(100,4), RM(100), BE(100), DBDM(100), SAL(100), AAA(100)

DIMENSION KKK(18)

DIMENSION W(2000), RWM(2000), BETAI(2000), WMB(100,4), WT(100,4),
1XDOWNW(800), YACROS(800)

EQUIVALENCE (A(1,1), W(1)), (A(1,2), RWM(1)), (A(1,3), BETAI(1)),
1(A(1,4), WMB(1)), (A(401,4), WT(1)), (A(801,4), XDOWNW(1)),
2(K(1), YACROS(1))

INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
1FIRST, UPPER, S1, ST, SRW

REAL K, KAK, LAMBOA, LMAX, MH, MLE, MR, MSP, MV, MVIM1

DATA KKK(4)/1H/, KKK(6)/1H/, KKK(8)/1H/, KKK(10)/1H/, KKK(12)/1H/, KKK(14)/1H/, KKK(16)/1H/, KKK(18)/1H/

CALL VELBB AND VELSUR THROUGHOUT THE REGION
MBOT = MINO(MBIIM1,MBOO)
IF (MBOT.LT.2) GO TO 20
DO 10 IM=2,MBOT
CALL VELBB(IM,5,2)
10 CALL VELBB(IM,1,6)
20 MBIT = MAXO(2,MBIT)
MBOT = MINO(MMM1,MBOO)
IF (MBIT.GT.MBOT) GO TO 40
DO 30 IM=MBIT,MBOT
CALL VELBB(IM,5,4)
CALL VELBB(IM,3,2)
30 CALL VELBB(IM,1,6)
GO TO 60
40 MBIT = MAXO(2,MBOOP1)
MBOT = MINO(MBIIM1,MMM1)
IF (MBIT.GT.MBOT) GO TO 60
DO 50 IM=MBIT,MBOT
50 CALL VELBB(IM,5,6)
60 MBIT = MAXO(MBIIM1,MBOOP1)
IF (MBIT.GT.MMM1) GO TO 80
DO 70 IM=MBIT,MMMI1
CALL VELBB(IM,5,4)
70 CALL VELBB(IM,3,6)
80 CALL VELSUR
C
C PREPARE INPUT ARRAYS FOR PLOT OF VELOCITIES
C
NP2 = 0
C SURFACES 1 TO 4 - TANGENTIAL COMPONENTS
DO 110 SURF=1,4
NP1 = NP2
IMSS = IMS(SURF)
IF (IMSS.LT.1) GO TO 100
DO 90 IHS=1,IMSS
IF (WTB(IHS,SURF).EQ.0.) GO TO 90
IF (ABS(DTMH(IHS,SURF)*RMH(IHS,SURF))*.LT..57735) GO TO 90
NP1 = NP1+1
YACROS(NP1) = WTB(IHS,SURF)
XDOWN(NP1) = MH(IHS,SURF)
90 CONTINUE
100 KKK(2*SURF+1) = NP1-NP2
110 NP2 = NP1
C SURFACES 1 AND 2 - MERIDIONAL COMPONENTS
DO 140 SURF=1,2
NP1 = NP2
MBOT = MINO(MBOOM1,MMMM1)
IF (2.GT.MBOT) GO TO 130
DO 120 IM=2,MBOT
IF (WMB(IM,SURF).EQ.0.) GO TO 120
IF (ABS(DTMV(IM,SURF)*RM(IM))*.GT.1.7321) GO TO 120
NP1 = NP1+1
YACROS(NP1) = WMB(IM,SURF)
XDOWN(NP1) = MV(IM)
120 CONTINUE
130 KKK(2*SURF+9) = NP1-NP2
140 NP2 = NP1
C SURFACES 3 AND 4 - MERIDIONAL COMPONENTS
DO 170 SURF=3,4
NP1 = NP2
MBIT = MAXO(MBIIP1,2)
IF(MBIT.GT.MMMM1) GO TO 160
DO 150 IM=MBITtMMM1
IF(WYB(IM,SURF).EQ.O.) GO TO 150
IF ( ABS ( DTnMV ( IM, SURF ) ) .GT. 1.732 ) GO TO 150
NP1= NP1+1
YACROS(NP1)= WMB(IM, SURF)
XDOW(NP1)= MV(IM)
150 CONTINUE
160 KKK(2*SURF+9) = NP1-NP2
170 NP2= NP1
C
C PLOT VELOCITIES
C
KKK(1)= 1
KKK(2)= 8
P= 5.
WRITE(6,1000)
CALL PLOTMY(XDOWN,YACROS,KKK,P)
WRITE(6,1010)
RETURN
1000 FORMAT(2HPT,50X,24HBLADE SURFACE VELOCITIES)
1010 FORMAT (2HPL,37X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
1(M) DOWN THE PAGE /2HPL/
22HPL,50X,50H+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT/
32HPL,50X,50H* - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT/
42HPL,50X,50HX - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT/
52HPL,50X,50HO - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT/
62HPL,50X,50H$ - BLADE SURFACE 3, BASED ON MERIDIONAL COMPONENT/
72HPL,50X,50H# - BLADE SURFACE 3, BASED ON TANGENTIAL COMPONENT/
82HPL,50X,50H) - BLADE SURFACE 4, BASED ON MERIDIONAL COMPONENT/
92HPL,50X,50H+ - BLADE SURFACE 4, BASED ON TANGENTIAL COMPONENT) 
END

SUBROUTINE VELBB(IM, UPPER, LOWER)
C
C VEL CALCULATES DENSITIES AND VELOCITIES FROM THE PRODUCT OF 
C DENSITY TIMES VELOCITY
C
COMMON SRW,ITER, IEND, LER(2), NER(1)
COMMON /INP/ GAM, AR, TIP, RHOIP, WTFL, WTFLSP, OMEGA, ORF, BETAI, BETAO,
1NOBL, MB1, MB2, MBI2, MM, N8BI, NBL, NRS, MBDYF, MBDYL, ITF, ITL,
2BLDAT, AANDK, ERSTOR, STRFN, INTVL, SURVL, MAGFAC,
3MR(50), RNSP(50), RESP(50)
COMMON /CALCON/ MB1, MB2, MM, MBI1, MB11, MB12, MB21, MM, N8BI, NBL, NRS,
1LAM0, TLW, ITOR, ITMAF, NIP, IMS(4), BV(4), MV(100), IV(101),
3UBV(100,4), RWBV(100,4), ITV(100,6), TV(100,4), DTMV(100,4),
4BETAIV(100,4), MH(100,4), DTMH(100,4), BETAH(100,4), RMH(100,4),
5BETH(100,4), RMM(100), RM(100), BE1(100), DBDM(100), SAL(100), AAA(100)
COMMON /RHS/ RHOHS(100,4), RHOVS(100,4), RVB(100,4)
DIMENSION WWCRM(100,4), WWCRT(100,4)
DIMENSION W(2000), RWM(2000), BETA(2000), WMB(100,4), WTB(100,4),
1XDOW(N800), YACROS(800)
EQUIVALENCE (A(1,1), W(1)), (A(1,2), RWMB(1)), (A(1,3), BETA(1)), 
1(A(1,4), WMB(1)), (A(401,4), WTB(1)), (A(801,4), XDOW(1)), 

68
VELBB calculates along vertical mesh lines from blade to blade

IF (IM. NE. 2. OR. UPPER. NE. 5) GO TO 10
IF (INTVL. GT. 0) WRITE(6,1000)
  TELER = 0.
10  ITVU = MAX(0(ITV(IM, UPPER), 2)
  ITVL = MINO(ITV(IM, LOWER), ITMAX-1)
  IPUP1 = IPF(IM, ITVU)
  IPLM1 = IPF(IM, ITVL)
  TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RH(M)(IM))**2
  WCR = SORT(TGROG*TIP*(1.-TWLMR/CPTIP))
  IF (ITVL. LT. ITVU) GO TO 30
IF (UPPER. EQo5) RETURN
IF (ITV(IM,LOWER) .GT. ITMAX-1) WMB(IM, LOWER) = 0.
  RHOB = RHOBV(IM, UPPER)
  LER(1) = 6
C DENSTY CALL NO. 6
CALL DENSTY(WMB(IM,UPPER), RHOB(IM,UPPER), ANS, TWLMR, CPTIP, EXPON, 1RHOB, GAM, AR, TIP)
  WMB(IM,UPPER) = ANS
  WWCMR(M,UPPER) = WMB(IM, UPPER)/WCR
  RELFR = AMAX1(RELFR, ABS((RHOB-RHOBV(IM,UPPER)) / RHOBV(IM,UPPER)))
C ON THE LOWER SURFACE, IF IT IS A BLADE
40  IF (LOWER. EQ. 6) RETURN
IF (ITV(IM,LOWER) .GT. ITMAX-1) WMB(IM, LOWER) = 0.
  RHOB = RHOBV(IM, LOWER)
  LER(1) = 7
C DENSTY CALL NO. 7
CALL DENSTY(WMB(IM,LOWER), RHOBV(IM, LOWER), ANS, TWLMR, CPTIP, EXPON, 1RHOB, GAM, AR, TIP)
  WMB(IM,LOWER) = ANS
  WWCRM(M,LOWER) = WMB(IM, LOWER)/WCR
  RELFR = AMAX1(RELFR, ABS((RHOB-RHOBV(IM, LOWER)) / RHOBV(IM, LOWER)))
RETURN
C VELSUR calculates along a blade surface
C
ENTRY VELSUR
DO 60 SURF=1,4
  IMSS = IMS(SURF)
  IF (IMSS. EQ. 0) GO TO 60
DO 50 IHS=1,IMSS
  TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RH(M)(IHS,SURF))**2
  WCR = SORT(TGROG*TIP*(1.-TWLMR/CPTIP))
  RHOB = RHOBV(IHS,SURF)
  LER(1) = 8
C DENSITY CALL NO. 8
CALL DENSITY(WTB(IHS,SURF),RHOH8(IHS,SURF),ANS,TLMR,CPTIP,EXPO,
RHOIP,GAM,AR,TIP)
WTB(IHS,SURF)= ANS
WWCRT(IHS,SURF)= WTB(IHS,SURF)/WCR
50 RELER= AMAX1(RELER,ABS((RHOB-RHOH8(IHS,SURF)) / RHOH8(IHS,SURF)))
60 CONTINUE
IF (RELER.LT..001) IEND=IEND+1
WRITE(6,1020) ITER,RELER
MARK = 0

C WRITE ALL BLADE SURFACE VELOCITIES
C
IF (SURVL.LE.0) RETURN
WRITE(6,1030)
MBOT = MIN0(MMML,MB0C)
IF(MBOT.LT.,2) GO TO 70
WRITE(6,1040)
WRITE(6,1050) (MV(IM),WMB(IM,1),BETAV(IM,1),WWCRM(IM,1),WMB(IM,2),
IBETAV(IM,2),WWCRM(IM,2),IM=2,MBOT)
70 MBIT = MAX0(2,MBIT)
IF(MBIT.GT.,MMML) GO TO 80
WRITE(6,1060)
WRITE(6,1050) (MV(IM),WMB(IM,3),BETAV(IM,3),WWCRM(IM,3),WMB(IM,4),
IBETAV(IM,4),WWCRM(IM,4),IM=MBIT,MMML)
80 WRITE(6,1070)
DO 90 SURF=1,4
IMSS = IMS(SURF)
IF (IMSS.EQ.0) GO TO 90
WRITE(6,1080) SURF
WRITE(6,1090) (MH(IHS,SURF),WTB(IHS,SURF),BETAH(IHS,SURF),WWCRT
1(IHS,SURF),IHS=1,IMSS)
90 CONTINUE
RETURN

1000 FORMAT(1H1//40X,34HVELOCITIES AT INTERIOR MESH POINTS//)
1010 FORMAT(1HL,3HIM=13,5(24H VELOCITY ANGLE(DEG))/
1(5X,5(G15.4,F9.2))
1020 FORMAT(14HLITERTATION NO.,I3,3X,36XMAXIMUM RELATIVE CHANGE IN DENS
ITY =,G11.4)
1030 FORMAT(1H1/16X,1H*,25X,49HSURFACE VELOCITIES BASED ON MERIDIONA
L COMPONENTS,36X,1H*)
1040 FORMAT(16X,1H*,53X,1H*,56X,1H*/16X,1H*,19X,15HBLADE SURFACE 1,19X,
11H*,20X,15HBLADE SURFACE 2,21X,1H*/7X,1HM,8X,1H*,2(3X,8HVELOCITY,13X,10HANGLE(DEG),5X,5HW/WCR,19X,1H*,3X))
1050 FORMAT(1H ,G13.4,F9.2,G15.4,G15.4,F9.2,G15.4,14,17X,1H*)
1060 FORMAT(///16X,1H*,19X,15HBLADE SURFACE 3,19X,1H*,20X,15HBLADE SURF
ACE 4,21X,1H*/7X,1HM,8X,1H*,2(3X,8HVELOCITY,3X,10HANGLE(DEG),5X,15H/W/WCR,19X,1H*,3X))
1070 FORMAT(1H1//3X,49HSURFACE VELOCITIES BASED ON TANGENTIAL COMPONENT
S)
1080 FORMAT(///22X,15HBLADE SURFACE 5,11/7X,1HM,10X,8HVELOCITY,3X,10HANH
GLE(DEG),3X,5HW/WCR)
1090 FORMAT(1H ,G13.4,F9.2,G15.4)
END
SUBROUTINE BLCD

BLCD CALCulates BLade THeta COORDINATE AS A FUNCTION OF M

COMMON SRW,ITER,IER,ER(2),NER(1)
COMMON /INP/ GAM,AR,TIP,RHOIP,WTFL,WTFSP,OMEGA,ORF,BETA1,BETA0,
1N0BL,MB1,MB0,MB12,MB02,MM,NBBI,NBL,NRSP,MBDYF,MBDLT,ITF,ITL,
2BLDAT,AANDK,ERSOR,STRFN,INTVL,SURVL,MAGFAC,
3MR(50),RMS(50),BES(50)
COMMON /CALCON/ MB11,MB00,MM,MBI1M1,MBI1P1,MB00M1,MB00P1,MMM1,
1HM1,HM2,HM3,HT,DTLR,DMLR,PITCH,CP,EXPON,TRM,TST,BIB,TIB,
2LAMBDA,TWL,ITR,ITMAX,NIP,MS(4),BV(4),MV100,IV(101),
3SUBV(100,4),RWBV(100,4),ITV(100,6),TV(100,4),DTDMV(100,4),
4BETA(100,4),MB100,4),DTDM(100,4),BETA(100,4),RM(100,4),
5BEH100,4),RM(100),BE(100),DBDM(100),SAL(100),AAA(100)
COMMON /GEOMIN/ CHORD(4),STGR(4),MLE(4),THLE(4),RMO(4),
1RI(4),RO(4),BETI(4),BETO(4),NSPI(4),MSP(50,4),THSP(50,4)
COMMON /BLDCM/ EM(50,4),INIT(4)
ENTRY BL1(M,THETA,DTDM,INF)
ENTRY BL2(M,THETA,DTDM,INF)
ENTRY BL3(M,THETA,DTDM,INF)
ENTRY BL4(M,THETA,DTDM,INF)

10 INF= 0
NSP= NSPI(SURF)
IF (INIT(SURF).EQ.13) GO TO 30
INIT(SURF)= 13

C INITIAL CALCULATION OF FIRST AND LAST SPLINE POINTS ON BLADE

AA = BETA(SURF)/57.295779
AA = SIN(AA)
MSP(1,SURF) = RI(SURF)*(1.-SIGN*AA)
BB = SQRT(1.-AA**2)
THSP(1,SURF) = SIGN*BB*RI(SURF)/RMI(SURF)
BETI(SURF) = AA/BB/RMI(SURF)
AA = BETO(SURF)/57.295779
AA = SIN(AA)
MSP(NSP,SURF) = CHORD(SURF)-RO(SURF)*(1.+SIGN*AA)
BB = SQRT(1.-AA**2)
THSP(NSP,SURF) = STGR(SURF)+SIGN*BB*RO(SURF)/RMO(SURF)
BETO(SURF) = AA/BB/RMO(SURF)
DO 20 IA=1,NSP
MSP(IA,SURF)= MSP(IA,SURF)+MLE(SURF)
20 THSP(IA,SURF)= THSP(IA,SURF)+THLE(SURF)
CALL SPLN22(MSP(1,SURF),THSP(1,SURF),BETI(SURF),BETO(SURF),NSP,
IAAA, EM(1, SURF))
IF (BLODAT LE.0) GO TO 30
IF (SRW EQ.0) WRITE(6, 1000)
SRW = 1
WRITE(6, 1010) SURF
WRITE (6, 1020) (MSP(IA, SURF), THSP(IA, SURF), AAA(IA), EM(IA, SURF),
I1A=1, NSP)
C
C BLADE COORDINATE CALCULATION
C
30 KK = 2
IF (M GT. MSP(1, SURF)) GO TO 50
C
AT LEADING EDGE RADIUS
C
MMLE = M - MLE(SURF)
IF (MMLE LT. -DMLR) GO TO 90
MMLE = AMAX1(0., MMLE)
THETA = SQRT(MMLE*(2.*RI(SURF) - MMLE)) *SIGN
IF (THETA EQ.0.) GO TO 40
RMM = RI(SURF) - MMLE
DTDM = RMM/THETA/RMI(SURF)
THETA = THETA/RMI(SURF) + THLE(SURF)
RETURN
40 INF = 1
DTDM = 1.E10 *SIGN
THETA = THLE(SURF)
RETURN
C
ALONG SPLINE CURVE
C
50 IF (M LE. MSP(KK, SURF)) GO TO 60
IF (KK GE. NSP) GO TO 70
KK = KK + 1
GO TO 50
60 S = MSP(KK, SURF) - MSP(KK-1, SURF)
EMKM1 = EM(KK-1, SURF)
EMK = EM(KK, SURF)
MMSPM = MSP(KK, SURF) - M
MMSP = M - MSP(KK-1, SURF)
THK = THSP(KK, SURF)/S
THKML = THSP(KK-1, SURF)/S
THETA = EMKM1*MSPM**3/6./S + EMK*MMSP**3/6./S + (THK- EMK*S/6.)*
1 MMSM + (THKM1-EMKM1*S/6.)*MSPM
DTDM = -EMKM1*MSPM**2/2./S + EMK*MMSP**2/2./S + THK-THKM1-(EMK-1 EMKM1)*S/6.
RETURN
C
AT TRAILING EDGE RADIUS
C
70 CMM = CHORD(SURF) + MLE(SURF) - M
IF (CMM LT. -DMLR) GO TO 90
CMM = AMAX1(0., CMM)
THETA = SQRT(CMM*(2.*RO(SURF) - CMM)) *SIGN
IF (THETA EQ.0.) GO TO 80
RMM = RO(SURF) - CMM
DTDM = -RMM/THETA/RMO(SURF)
THETA = STGR(SURF) + THETA/RMO(SURF) + THLE(SURF)
RETURN
80 INF = 1
DTDM = -1.0E10*SIGN
THETA= THLE(SURF)+STGR(SURF)
RETURN
C
C  ERROR RETURN
C
90  WRITE (6,1030) LER(2),M,SURF
STOP
1000 FORMAT (1H1,13X,27HBLADE DATA AT SPLINE POINTS)
1010 FORMAT (1HL,17X,16HBLADE SURFACE,I4)
1020 FORMAT (7X~1HM~1OX~5HTHETA~1OX~1OHDERIVATIVE~5X~10H2ND DERIV.~/
14G15.5))
1030 FORMAT (14HLBLCD CALL NO.,I3/33H M COORDINATE IS NOT WITHIN BLADE/
14H M =,G14.6,10X,6HSURF =,G14.6)
END

FUNCTION IPF(IM,IT)
COMMON /CALCOC/ MBI,MB00,MMM,MBIIM1,MBIIP1,MBOOM1,MBOOP1,MMM1,
1HM1,HM2,HM3,HT,DLR,DLR,PITCH,CP,EXPON,TWN,CTIP,TGROG,TBI,TBO,
2LAMBDA,TLR,ITOR,ITMAX,NI,IM5(4),BV(4),MV(100),IV(101),
3UBV(100,4),RMVBV(100,4),ITV(100,6),TV(100,4),DTDMV(100,4),
4BETA(100,4),MHP(100,4),DTS(100,4),BETAH(100,4),RMH(100,4),
5BEP(100,4),RMP(100),BEG(100),DBDM(100),SAL(100),AAA(100)
IPF= IV(IM)+1T-2
IF(IT.LT.1TIV(IM,4)) RETURN
IF(IT.LT.1TIV(IM,2) .AND. IM.LT.MBI) RETURN
IF(IM.GE.MINO(MBI,MB00)) GO TO 10
IF (ITV(IM,1).EQ.-10000) RETURN
IF(IPF-ITV(IM,1) + ITV(IM,2)+1
RETURN
10 IF(IM.GT.MAXO(MBI,MBO0)) GO TO 20
IF(IM.GT.MBOO) RETURN
IPF= IPF-ITV(IM,3)+ITV(IM,4)+1
IF(IT.LT.1TIV(IM,1)) RETURN
IPF= IPF-ITV(IM,1)+ITV(IM,2)+1
RETURN
20 IF (ITV(IM,3).EQ.-10000) RETURN
IPF= IPF-ITV(IM,3)+ITV(IM,4)+1
RETURN
END

Subroutine BDVINT

BDVINT calculates interpolated values of the stream function along either a vertical or a horizontal boundary for the fine mesh. The interpolation is based on a cubic spline curve (ref. 8) using the stream-function values at the original coarse-mesh points.

The input arguments for BDVINT are as follows:

BVIN   input array from main program (see DESCRIPTION OF INPUT AND OUTPUT)
UBVIN input array from main program (see DESCRIPTION OF INPUT AND OUTPUT)

NSP given number of stream-function values on the coarse mesh

MVTH array of either $m(MV)$ or $\theta(TH)$ coordinates for the fine mesh

KBDRY input variable (see DESCRIPTION OF INPUT AND OUTPUT)

DLR tolerance, either $DMLR$ or $DTLR$

MMMITX total number of fine mesh lines, either $MMM$ or $ITMAX$

The internal variables for BDVINT are as follows:

**BVINT** array of values of $m$ or $\theta$ where interpolated stream-function values are desired

**IM** index of mesh line

**IM1** value of IM for first point in region

**IM2** $IM1 + 1$

**IMT** value of IM for last point in region

**NSPINT** number of interpolated values of stream function

**RWBV** array of interpolated values of either $\partial u/\partial m$ or $\partial u/\partial \theta$

**UBV** array of interpolated stream-function values

SUBROUTINE BDVINT(BVIN,UBVIN,NSP,MVTH,KBDRY,DLR,MMMITX)
COMMON /CALCON/ M8II,MBOO,MMH,M8IIIM1,M8IIPL,MBOOM1,MBOOPL,MMM1,
1HM1,HM2,HM3,HT,DLR,DMLR,PITCH,CP,EXPON,TW,CTIP,TGROG,TBI,TBO,
2ALAMDA,TWL,ITOR,ITMAX,NIP,IMS(4),BV(4),MV(100),IV(101),
3UBV(100,4),RWBV(100,4),ITV(100,6),TV(100,4),DTRMV(100,4),
4BETAV(100,4),MH(100,4),DTRM(100,4),BEH(100,4),RMH(100,4),
5BEH(100,4),RMH(100,4),BEL(100),DBDM(100),SAL(100),AAA(100),
DIMENSION BVINT(100),BVIN(100),UBVIN(100),MVTH(100)
REAL MVTH
DO 10 IM=2,MMMITX
IF (MVTH(IM).GT.BVIN(NSP)+DLR) GO TO 20
10 CONTINUE
IM = MMMITX
20 IM1 = IM-1
BVINT(IM1) = BVIN(1)
IM2 = IM
DO 30 IM=IM2,MMMITX
IF (MVTH(IM).GT.BVIN(NSP)-DLR) GO TO 40
30 BVINT(IM) = MVTH(IM)
IM = MMMITX
40 IMT=IM
BVINT(IMT) = BVIN(NSP)
NSPINT= IMT-IM1+1
CALL SPLINT(BVIN,UBVIN,NSP,BVINT(IM1),NSPINT,UBV(IM1,KBDRY),RWBV(IM1,KBDRY))
RETURN
END
Program Listing of Remaining Subroutines

The remaining subroutines are described in reference 3. The description applies even though the subroutines have been revised.

SUBROUTINE MHORIZ(MV, ITV, BL, MBI, MBO, ITO, HT, DTLR, KODE, J, MH, DTDMH, IMRTS)

C
C MHORIZ CALCULATES M COORDINATES OF INTERSECTIONS OF ALL HORIZONTAL
C MESH LINES WITH A BLADE SURFACE
C KODE = 0 FOR UPPER BLADE SURFACE
C KODE = 1 FOR LOWER BLADE SURFACE
C
COMMON SRW, ITER, IEND, LER(2), NER(1)
DIMENSION MV(100), ITV(100), MH(100), DTDMH(100)
INTEGER BLDAT, AANDK, ERSOR, STRFN, SURVL, AATEMP, SURF,
IFIRST, UPPER, S1, ST, SRW
REAL K, KAK, LAMBD, LMAX, MH, MLE, MR, MSP, MV, MVIM
REAL MVIM
EXTERNAL BL
IF (MBI.GE.MBO) RETURN
IM = MBI
10 IMIND = 0
20 IF (ITV(IM+1)-ITV(IM)-ITIND) 30, 40, 50
30 J = J+1
   TI = FLOAT(ITV(IM+1)-ITO-ITIND+KODE)*HT
   ITIND = ITIND-1
   MVIM = MV(IM)
   IF (MRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
   CALL ROOT(MVIM, MV(IM+1), TI, BL, DTLR, MH(J), DTDMH(J))
   GO TO 20
40 IM = IM+1
   MRTS = 0
   IF (IM.EQ.MBO) RETURN
   GO TO 10
50 J = J+1
   TI = FLOAT(ITV(IM)-ITO+ITIND+KODE)*HT
   ITIND = ITIND+1
   MVIM = MV(IM)
   IF (MRTS.EQ.1) MVIM = MVIM+(MV(IM+1)-MVIM)/1000.
   CALL ROOT(MVIM, MV(IM+1), TI, BL, DTLR, MH(J), DTDMH(J))
   GO TO 20
END
SUBROUTINE DENSITY(RHOW,RHO,VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
DENSITY CALCULATES DENSITY AND VELOCITY FROM THE WEIGHT FLOW PARAMETER
DENSITY TIMES VELOCITY

COMMON SRW,ITER,IEND,LER(2),NER(1)
VEL = HOW/RHO
IF (VEL.NE.0.) GO TO 10
RHO = RHOIP
RETURN
10 TTIP = 1.-(VEL**2+TWLMR)/CPTIP
IF (TTIP.LT.0.) GO TO 30
TEMP = TTIP**(EXPON-1.)
RHOIP = RHOIP*TEMP*TTIP
RHOWP = -VEL**2/GAM*RHOIP/AR*TEMP/TIP+RHO
IF (RHOWP.LE.0.) GO TO 30
VELNEW = VEL+(RHOW-RHOIP*VEL)/RHOWP
IF (ABS(VELNEW-VEL)/VELNEW.LT.0.0001) GO TO 20
VEL = VELNEW
GO TO 10
20 VEL = VELNEW
RHO = HOW/VEL
RETURN
30 TGROG = 2.*GAM*AR/(GAM+1.)
VEL = SQRT(TGROG*CPTIP*(1.-TWLMR/CPTIP))
RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
RWHORW = RHOW/VEL
NER(1) = NER(I)+1
WRITE (6,1000) LER(1),NER(1),RWHORW
IF (NER(1).LT.1000) STOP
RETURN
1000 FORMAT (16HL,DENSITY CALL NO.13/H,NER(1) =,13/10H RWHORW IS ,F7.4,
134H TIMES THE MAXIMUM VALUE FOR RWHORW)
END

SUBROUTINE ROOT(A,B,Y,FUNCT,TOLERY,X,DFX)
ROOT FINDS A ROOT FOR (FUNCT MINUS Y) IN THE INTERVAL (A,B)

COMMON SRW,ITER,IEND,LER(2),NER(1)
INTEGER SRW
IF (SRW.EQ.21) WRITE(6,1000) A,B,Y,TOLERY
TOLERY = (B-A)/1000.
AB2 = (A+B)/2.
I= 0
X= A
10 CALL FUNCT(X,FX,DFX,INF)
IF (SRW.EQ.21) WRITE(6,1010) I,X,FX,DFX,INF
IF (ABS(Y-FX).LT.TOLERY) RETURN
IF (I.GE.1000) GO TO 30
I= I+1
IF (INF.NE.0 .OR. DFX.EQ.0.) GO TO 20
X= (Y-FX)/DFX+X
IF (X.GE.A .AND. X.LE.B) GO TO 10
X = A+TOLERY*FLOAT(I)
76
IF(I.EQ.1) X = B
GO TO 10
20 IF (X.LT.AH2) X=X+TOLERX
IF (X.GE.oAH2) X=X-TOLERX
GO TO 10
30 WRITE(6,1020) LER(2),A,B,Y
STOP
1000 FORMAT (32HINPUT ARGUMENTS FOR ROOT -- A =G13.5,3X,3HB =,G13.5,
13X,3HY =,G13.5,3X,8HTOLERY =,G13.5/17H ITER. NO. X,17X,
22HFX,15X,3HDFX,10X,3HINF)
1010 FORMAT (5X,I3,G16.5,2G18.5,I6)
1020 FORMAT (14HLROOT CALL NO.,I3/47H ROOT HAS FAILED TO CONVERGE IN 10
100 ITERATIONS/4H A =,G14.6,10X,3HB =,G14.6,10X,3HY =,G14.6)
END

SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
C
C SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
C ADJACENT POINT
C
COMMON Q/B0X/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
1G(200)
DIMENSION X(N),Y(N),EM(N),SLOPE(N)
INTEGER Q
DO 10 I=2,N
10 S(I)=X(I)-X(I-1)
NO=N-1
IF(NO.LT.2) GO TO 30
DO 20 I=2,NO
A(I)=S(I)/6.
B(I)=(S(I)+S(I+1))/3.
C(I)=S(I+1)/6.
20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
30 A(N) = -.5
B(I)=1.
B(N)=1.
C(I) = -.5
F(I)=0.
F(N)=0.
W(I)=B(I)
SB(I)=C(I)/W(I)
G(I)=0.
DO 40 I=2,N
W(I)=B(I)-A(I)*SB(I-1)
SB(I)=C(I)/W(I)
40 G(I)=(F(I)-A(I)*G(I-1))/W(I)
EM(N)=G(N)
DO 50 I=2,N
K=N+1-I
50 EM(K)=G(K)-SB(K)*EM(K+1)
SLOPE(I)=-S(2)/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/S(2)
DO 60 I=2,N
60 SLOPE(I)=S(I)/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/S(I)
IF (Q.EQ.13) WRITE (6,1000) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
RETURN
1000 FORMAT (2X,15HND. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
12HEM/(4F20.8))
END

SUBROUTINE SPLN22 (X,Y,Y1P,YNP,N,SLOPE,EM)
C
C SPLN22 CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C END CONDITION - DERIVATIVES SPECIFIED AT END POINTS
C
COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
1G(200)
DIMENSION X(N),Y(N),EM(N),SLOPE(N)
INTEGER Q
DO 10 I=2,N
10 S(I)=X(I)-X(I-1)
NO=N-1
IF(NO.LT.2) GO TO 30
DO 20 I=2,NO
A(I)=S(I)/6.
B(I)=(S(I)+S(I+1))/3.
C(I)=S(I+1)/6.
20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
30 A(N) = S(N)/6.
B(N) = S(N)/3.
C(I)=S(2)/6.
F(1)=(Y(2)-Y(1))/S(2)-Y1P
F(N) = YNP-(Y(N)-Y(N-1))/S(N)
W(1)=B(1)
SB(1)=C(1)/W(1)
G(1)=F(1)/W(1)
DO 40 I=2,N
W(I)=B(I)-A(I)*SB(I-1)
SB(I)=C(I)/W(I)
40 G(I)=(F(I)-A(I)*G(I-1))/W(I)
EM(N)=G(N)
DO 50 I=2,N
K=N+1-I
50 EM(K)=G(K)-SB(K)*EM(K+1)
SLOPE(I)=-S(2)/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/S(2)
DO 60 I=2,N
60 SLOPE(I)=S(I)/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/S(I)
IF (Q.EQ.18) WRITE (6,1000) N,(X(I),Y(I),SLOPE(I),EM(I),I=1,N)
RETURN
1000 FORMAT (2X,15HND. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
12HEM/(4F20.8))
END

78
SUBROUTINE SPLINT (X, Y, N, Z, MAX, YINT, DYDX)

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

COMMON Q/BOX/S100, A100, B100, C100, F100, W100, SB100,
1G100, EM100
DIMENSION X(N), Y(N), Z(MAX), YINT(MAX), DYDX(MAX)
INTEGER Q
IF(MAX.LE.0) RETURN
III = Q
DO 10 I=2, N
10 S(I)=X(I)-X(I-1)
NO=N-1
IF(NO.LT.2) GO TO 30
DO 20 I=2, NO
A(I)=S(I)/6.0
B(I)=(S(I)+S(I+1))/3.0
C(I)=S(I+1)/6.0
20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
30 A(N) = -.5
B(N)=1.0
C(N) = -.5
F(N)=0.0
W(N)=0.0
W(I)=B(I)
SB(I)=C(I)/W(I)
G(I)=0.0
DO 40 I=2, N
W(I)=B(I)-A(I)*SB(I-1)
SB(I)=C(I)/W(I)
40 G(I)=(F(I)-A(I)*G(I-1))/W(I)
EM(N)=G(N)
DO 50 I=2, N
K=N+1-I
50 EM(K)=G(K)-SB(K)*EM(K+1)
DO 140 I=1, MAX
K=2
IF(Z(I)-X(I)) 70, 60, 90
60 YINT(I)=Y(I)
GO TO 130
70 IF(Z(I).GE.(1.1*X(I)-.1*X(2))) GO TO 120
WRITE (6,1000) Z(I)
Q = 16
GO TO 120
80 K=N
IF(Z(I).LE.(1.1*X(N)-.1*X(N-1))) GO TO 120
WRITE (6,1000) Z(I)
Q = 16
GO TO 120
90 IF(Z(I)-X(K)) 120, 100, 110
100 YINT(I)=Y(K)
   GO TO 130
110 K=K+1
   IF(K-N) 90,90,80
120 YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6./S(K)+EM(K)*(-Z(I)-X(K-1))**3/6.
   1/S(K)+2*(Y(K)/S(K)-EM(K)*S(K)/6.)*(Z(I)-X(K-1))+2*(Y(K-1)/S(K)-EM(K-1)
   2*S(K)/6.)*(X(K)-Z(I))
130 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/S(K)+EM(K)*(-X(K-1)-Z(I))**2/2.
   10/S(K)+(Y(K)-Y(K-1))/S(K)-(EM(K)-EM(K-1))*S(K)/6.0
140 CONTINUE
   MXA = MAXO(N,MAX)
   IF(Q.EQ.16) WRITE(6,1010) N,MAX,X(I),Y(I),Z(I),YINT(I),DYDX(I),
   1I=1,MXA
   Q = III
   RETURN
1000 FORMAT (54H SPLINT USED FOR EXTRAPOLATION. EXTRAPOLATED VALUE = ,
   1G14.6)
1010 FORMAT (2X,2LHNO. OF POINTS GIVEN =,I3,30H, NO. OF INTERPOLATED PO
   IENTS =,I3/10X,1HX,19X,1HY,16X,11HX-INTERPOL.,9X,11HY-INTERPOL.,
   20X,14HDYDX-INTERPOL./15E20.8))
END

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, December 12, 1968,
126-15-02-31-22.

REFERENCES


"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— National Aeronautics and Space Act of 1958

**NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS**

**TECHNICAL REPORTS:** Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

**TECHNICAL NOTES:** Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

**TECHNICAL MEMORANDUMS:** Information receiving limited distribution because of preliminary data, security classification, or other reasons.

**CONTRACTOR REPORTS:** Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

**TECHNICAL TRANSLATIONS:** Information published in a foreign language considered to merit NASA distribution in English.

**SPECIAL PUBLICATIONS:** Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

**TECHNOLOGY UTILIZATION PUBLICATIONS:** Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

**SCIENTIFIC AND TECHNICAL INFORMATION DIVISION**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

Washington, D.C. 20546