NASA TECHNICAL MEMORANDUM

NASA TM X-1764

NASA TM X-1764

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

by Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1969

NASA TM X-1764

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

By Theodore Katsanis and William D. McNally

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

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

ABSTRACT

An existing FORTRAN IV computer program for blade-to-blade aerodynamic analysis of turbomachine blades was revised to obtain a simpler program consistent with related programs. The analysis is for two-dimensional, subsonic, compressible (or incompressible), nonviscous flow in a circular or straight infinite cascade of blades, which may be fixed or rotating. The flow may be axial, radial, or mixed, and the stream channel thickness may change in the through-flow direction. The results include streamline coordinates, velocity magnitude and direction throughout the passage, and the blade-surface velocities. This report includes a complete description of the input required by the program and the program listing.

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

by Theodore Katsanis and William D. McNally

Lewis Research Center

SUMMARY

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

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

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

INTRODUCTION

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

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

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

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

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

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

SYMBOLS

m	meridional streamline distance, meters, see figs. 1 and 2
r	radius from axis of rotation, meters
s	angular blade spacing or pitch, rad
V_θ	tangential component of absolute fluid velocity, meters/sec
W	fluid velocity relative to blade, meters/sec
z	axial coordinate, meters
α	angle between meridional streamline and axis of rotation, rad, see fig. 1
β	angle between relative velocity vector and meridional plane, rad, see fig. 1

θ relative angular coordinate, rad, see fig. 1

λ prerotation $\left(rV_{\theta}\right)_{in}$, meters^2/sec

ρ density, kg/meter^3

ω rotational speed, rad/sec

Subscripts:

cr critical velocity

in inlet or upstream

le leading edge

out outlet or downstream

te trailing edge

DESCRIPTION OF INPUT AND OUTPUT

The computer program requires as input a geometrical description in m-θ coordinates of the blade surfaces, a description in m-r coordinates of the stream channel

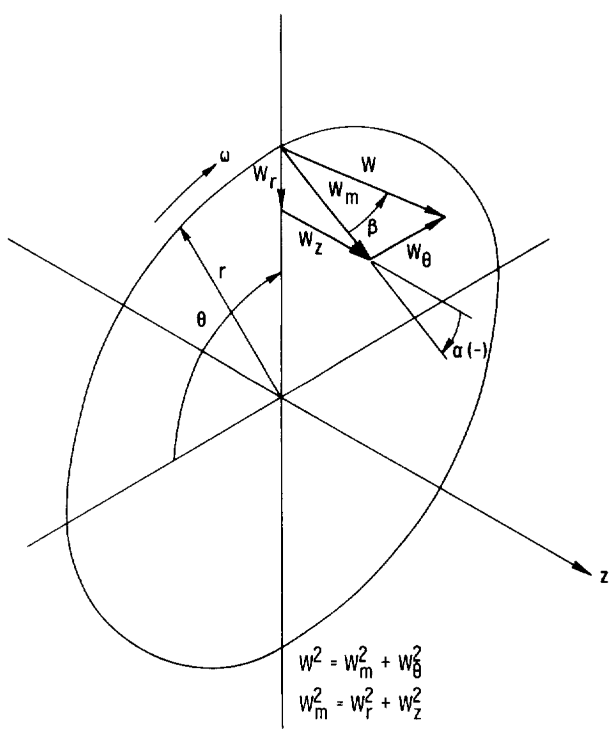


Figure 1. - Cylindrical coordinate system and velocity components.

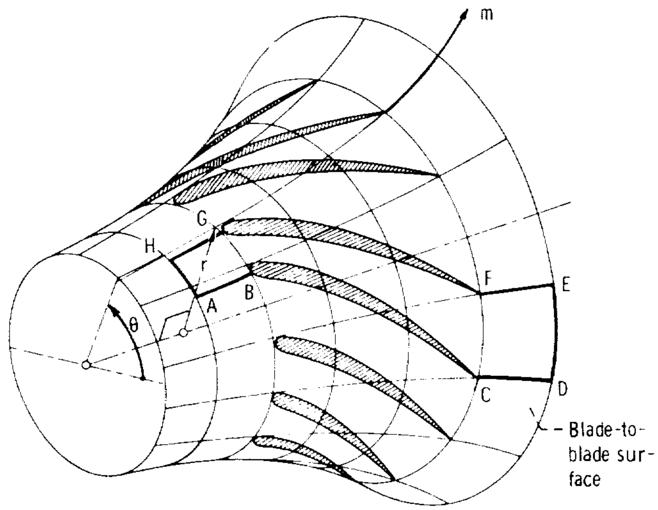


Figure 2. - Blade-to-blade surface of revolution.

1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40	41 50	51 60	61 70	71 80
TITLE											
GAM		AR		TIP		RHOIP		WTFL		OMEGA	ORF
BETAI		BETAO		CHORD		STGR					
MBI	MBO			MM	NBBI	NBL	NRSP				
RI1		RO1		BETI1		BETO1		SPLNO1			
MSP1 ARRAY											
THSP1 ARRAY											
RI2		RO2		BETI2		BETO2		SPLNO2			
MSP2 ARRAY											
THSP2 ARRAY											
MR ARRAY											
RMSP ARRAY											
BESP ARRAY											
BLDAT	ANNDK	ERSOR	STRFN	SLCRD	INTVL	SURVL					

Figure 3. - Input form. Card column numbers appear at top.

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

Input

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

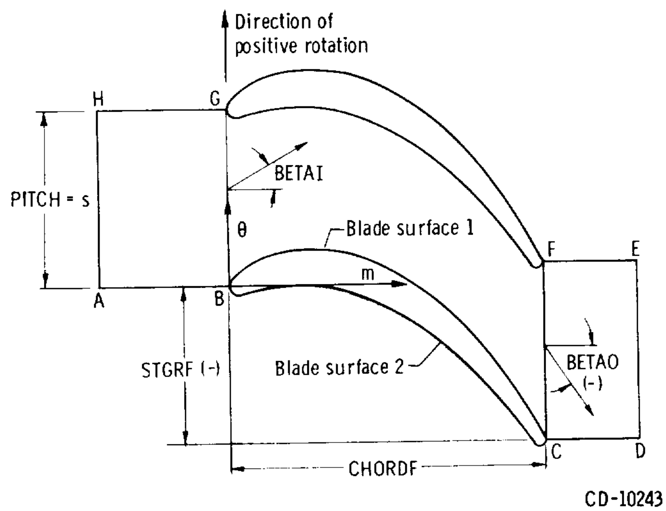


Figure 4. - Geometric input variables on blade-to-blade solution region.

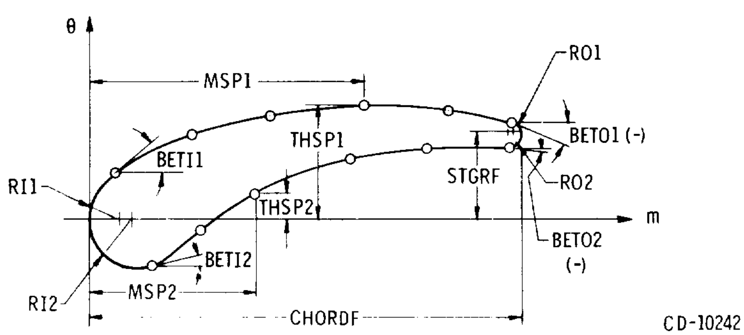


Figure 5. - Geometric input variables on a blade. BETI and BETO angles must be given as true angle β, not as angles measured in m- θ plane. Use $\tan\beta = r\,d\theta/dm$ to obtain β, or measure true angle.

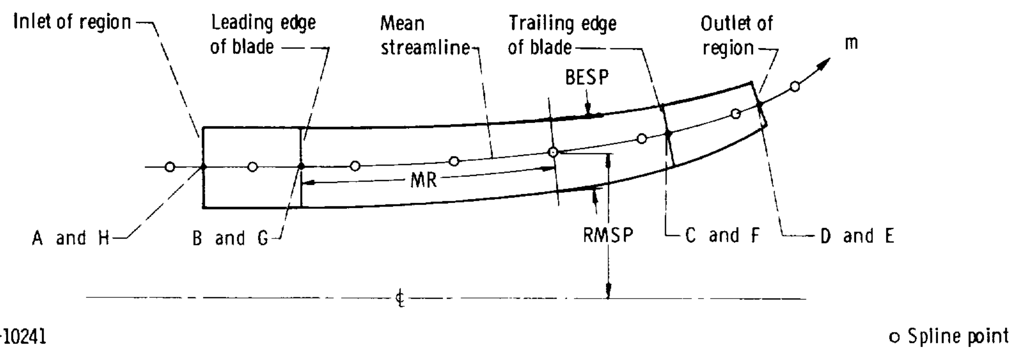


Figure 6. - Geometric input variables describing stream channel in meridional plane.

are shown in figures 4 to 6. All input variables are described in the section which follows. Further explanation of key variables is given in the section Instructions for Preparing Input.

The input variables are as follows:

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

MBO	number of vertical mesh lines from AH to CF inclusive (fig. 4)
MM	total number of vertical mesh lines in the m-direction from AH to DE, maximum of 100 (fig. 4)
NBBI	number of mesh spaces in θ-direction between AB and GH, maximum of 50 (fig. 4)
NBL	number of blades
NRSP	number of spline points for stream channel radius (RMSP) and thickness (BESP) coordinates, maximum of 50 (fig. 6)
RI1, RI2	leading-edge radii of the two blade surfaces, meters (fig. 5)
RO1, RO2	trailing-edge radii of the two blade surfaces, meters (fig. 5)
BETI1, BETI2	angles (with respect to m-direction) at tangent points of leading-edge radii with the two blade surfaces, deg (fig. 5) (These must be true angles in degrees. If angles are measured in the m-θ plane, i.e. $d\theta/dm$, BETI1 and BETI2 can be obtained from the relation $\tan \beta = r(d\theta/dm)$.)
BETO1, BETO2	angles (with respect to m-direction) at tangent points of trailing-edge radii with the two blade surfaces, deg (fig. 5) (These must also be true angles in degrees, like BETI1 and BETI2.)
SPLNO1, SPLNO2	number of blade spline points given for each surface as input, maximum of 50 (These include the first and last points (dummies) that are tangent to the leading- and trailing-edge radii (fig. 5).)
MSP1, MSP2	arrays of m-coordinates of spline points on the two blade surfaces, measured from the blade leading edge, meters (fig. 5) (The first and last points in each of these arrays can be blank or have a dummy value, since these points are calculated by the program. If blanks are used, and the last point is on a new card, a blank card must be used.)
THSP1, THSP2	arrays of θ-coordinates of spline points corresponding to MSP1 and MSP2, rad (fig. 5) (Dummy values are also used here in positions corresponding to those in MSP1 and MSP2.)
MR	array of m-coordinates of spline points for the stream channel radii and the stream channel thicknesses, meters (fig. 6) (MR is measured from the leading edge of the blade. These coordinates should cover the entire distance from AH to DE, and may extend beyond these bounds. The total number of points is NRSP.)

RMSP array of r-coordinates of spline points for the stream channel radii, corresponding to the MR array, meters (fig. 6)

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

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

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

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

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

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

SLCRD streamline θ-coordinates at each vertical mesh line, and streamline plot

INTVL velocity and flow angle at each iterior mesh point

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

Instructions for Preparing Input

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

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

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

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

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

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

Defining the mesh. - A finite-difference mesh is used for the solution of the basic differential equation. A typical mesh pattern is shown in figure 7. The mesh spacing and the extent of the upstream and downstream regions are determined by the values of MBI,

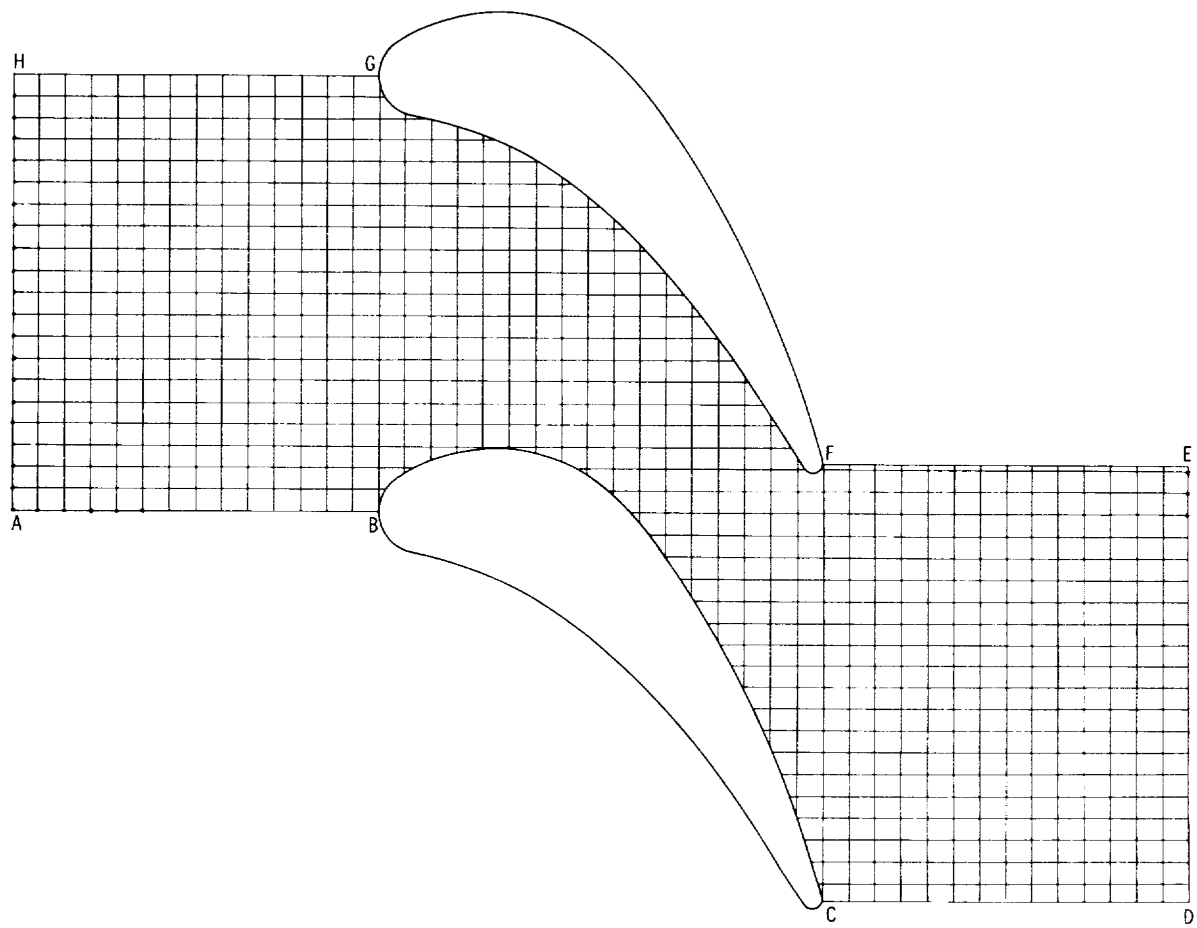


Figure 7. - Typical mesh in blade-to-blade solution region.

MBO, and MM of the input. The mesh spacing must be chosen so that there are not more than 2500 unknown mesh points.

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

Overrelaxation factor. - ORF is the overrelaxation factor used in each inner iteration in the solution of the simultaneous finite difference equations. (See ref. 2, p. 101).

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

Actually, the value of ORF is not as critical as the user might think. It gets more critical as the optimum value gets close to 2. For any run of a given set of data, only small changes will occur in the rate of convergence in SOR as long as the difference 2.0 - ORF is within 10 percent of its optimum value.

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

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

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

$$r = \frac{NBL*(sr)}{2\pi}$$

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

Axial flow. - For a two-dimensional cascade with constant stream channel thickness, constant values should be given for the MR, RMSP, and BESP arrays. Only two points are required for each of these arrays in this case. The two values of MR should be chosen so that they are further upstream and downstream than the boundaries AH and DE. The two values of RMSP and BESP should equal the constants r and b.

TABLE I. - SAMPLE OUTPUT

1

```
  AXIAL STATOR - MEAN SECTION - COMPRESSIBLE
   GAM              AR               TIP              RHOIP            WTFL             WTFLSP           OMEGA            ORF
 1.4000000        287.05300        288.15000        1.2250000        0.3146000        -0               -0               -0
  BETAI             BETAO            CHORDF           STGRF
 0               -67.000000        0.4265000E-01   -0.1116150
MBI  MBO                MM  NBBI  NBL NRSP
 15   32   -0   -0    47   20   50    2


 BLADE SURFACE 1  --  UPPER SURFACE
   RI1              RO1              BETI1            BETO1            SPLNO1
 0.3810000E-02    0.8890000E-03    28.300000       -72.400000        7.0000000
   MSP1  ARRAY
-0                0.8575000E-02    0.1715000E-01    0.2572500E-01    0.3430000E-01    0.3858800E-01   -0
   THSP1  ARRAY
-0                0.1769000E-01    0.1538000E-01   -0.5310000E-02   -0.4654000E-01   -0.7400000E-01   -0


 BLADE SURFACE 2  --  LOWER SURFACE
   RI2              RO2              BETI2            BETO2            SPLNO2
 0.3810000E-02    0.8890000E-03   -14.200000       -56.100000        6.0000000
   MSP2  ARRAY
-0                0.8575000E-02    0.1715000E-01    0.2572500E-01    0.3430000E-01   -0
   THSP2  ARRAY
-0               -0.1562000E-01   -0.2854000E-01   -0.5070000E-01   -0.8250000E-01   -0


   MR  ARRAY
-0.5000000E-01    0.1000000
   RMSP  ARRAY
 0.3302000        0.3302000
   BESP  ARRAY
 0.1016000        0.1016000

 BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
   1      2      2      2      2      2      3
```

2

```
            BLADE DATA AT INPUT SPLINE POINTS


                  BLADE     SURFACE   1
    M             THETA           DERIVATIVE       2ND DERIV.
 0.20037E-02     0.10159E-01      1.63066         -129.936
 0.85750E-02     0.17690E-01      0.60358         -182.661
 0.17150E-01     0.15380E-01     -1.23217         -245.503
 0.25725E-01    -0.53100E-02     -3.72155         -335.112
 0.34300E-01    -0.46540E-01     -5.54459         -90.0848
 0.38588E-01    -0.74000E-01     -7.92944         -1022.25
 0.42608E-01    -0.11080         -9.54694          217.603


                  BLADE     SURFACE   2
    M             THETA           DERIVATIVE       2ND DERIV.
 0.28754E-02     0.11448         -0.76632          57.4409
 0.85750E-02     0.11004         -0.96494         -127.138
 0.17150E-01     0.97124E-01     -2.04512         -124.799
 0.25725E-01     0.74964E-01     -3.12744         -127.637
 0.34300E-01     0.43164E-01     -4.32324         -151.265
 0.41023E-01     0.12547E-01     -4.50684          96.6479
```

3

	FREESTREAM VELOCITY	MAXIMUM VALUE FOR RHO*W	CRITICAL VELOCITY
LEADING EDGE B-G	61.9384	241.239	310.645
TRAILING EDGE C-F	180.067	241.239	310.645

	BETA CORRECTED TO BOUNDARY
BOUNDARY A-H	0
BOUNDARY D-E	-67.0000

4

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
0.1256637	0.6283185E-02	0.2508824E-02

ITMIN	ITMAX
-17	19

LAMBDA
0

5 NUMBER OF INTERIOR MESH POINTS = 846

6

SURFACE BOUNDARY VALUES

SURFACE	BV
1	0.
2	1.00000

2

BLADE DATA AT INTERSECTIONS OF VERTICAL MESH LINES WITH BLADES

	BLADE SURFACE 1		BLADE SURFACE 2	
M	TV	DTDMV	TV	DTDMV
0	0	0.10000E 11	0.12566	-0.10000E 11
0.25088E-02	0.10966E-01	1.56401	0.11482	-1.10043
0.50176E-02	0.14447E-01	1.20260	0.11291	-0.71758
0.75265E-02	0.16958E-01	0.79070	0.11099	-0.84944
0.10035E-01	0.18373E-01	0.32903	0.10850	-1.15031

2

STREAM SHEET COORDINATES AND THICKNESS TABLE

IM	M	R	SAL	B	DB/DM
1	-0.35124E-01	0.33020	-0	0.10160	-0
2	-0.32615E-01	0.33020	-0	0.10160	-0
3	-0.30106E-01	0.33020	-0	0.10160	-0
4	-0.27597E-01	0.33020	-0	0.10160	-0
5	-0.25088E-01	0.33020	-0	0.10160	-0

2

IM	IV ARRAY	ITV ARRAY	
	BLADE SURFACE NO.	1	2
1	1	0	19
2	21	0	19
3	41	0	19
4	61	0	19
5	81	0	19

TABLE I. - Continued. SAMPLE OUTPUT

2

M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINES WITH BLADE

MH ARRAY - BLADE SURFACE 1

MH	RMH	BEH	BETAH	DTDMH
0	0.3302	0.1016	90.000	0.1000E 11
0.6132E-03	0.3302	0.1016	57.041	4.6708
0.3582E-02	0.3302	0.1016	25.052	1.4156
0.1906E-01	0.3302	0.1016	-29.610	-1.7211
0.2201E-01	0.3302	0.1016	-40.095	-2.5498
0.2420E-01	0.3302	0.1016	-46.766	-3.2212

2

MH ARRAY - BLADE SURFACE 2

MH	RMH	BEH	BETAH	DTDMH
0.6132E-03	0.3302	0.1016	-57.041	-4.6708
0.4759E-02	0.3302	0.1016	-13.294	-0.7156
0.1140E-01	0.3302	0.1016	-23.596	-1.3228
0.1539E-01	0.3302	0.1016	-31.073	-1.8250
0.1850E-01	0.3302	0.1016	-36.169	-2.2140

2

THETA COORDINATES OF HORIZONTAL MESH LINES

IT	THETA
-17	-0.10681
-16	-0.10053
-15	-0.94248E-01
-14	-0.87965E-01
-13	-0.81681E-01

7

IT	IP	IP1	IP2	IP3	IP4	A(1)	A(2)	A(3)	A(4)	K
IM =	1	ITI =	0							
0	1	20	2	-0	21	0.	0.	0.	1.00000	0.
1	2	1	3	1	22	0.	0.	0.	1.00000	0.
2	3	2	4	2	23	0.	0.	0.	1.00000	0.
3	4	3	5	3	24	0.	0.	0.	1.00000	0.
4	5	4	6	4	25	0.	0.	0.	1.00000	0.

8
ESTIMATED OPTIMUM ORF = 2.000000
ESTIMATED OPTIMUM ORF = 1.999756
ESTIMATED OPTIMUM ORF = 1.999655
ESTIMATED OPTIMUM ORF = 1.999655
ESTIMATED OPTIMUM ORF = 1.999655
ESTIMATED OPTIMUM ORF = 1.999655

9
ERROR = 1.72602609
ERROR = 1.84942096
ERROR = 1.58245170
ERROR = 1.51459141
ERROR = 1.53313121

10

STREAM FUNCTION VALUES

IM = 1 IT1 = 0

0.02722913	0.07706153	0.12694010	0.17687678	0.22687771	0.27694251	0.32706463	0.37723198	0.42742761	0.47763271
0.52782747	0.57799304	0.62811383	0.67817773	0.72817850	0.77811576	0.82799505	0.87782843	0.92763250	0.97742604

IM = 2 IT1 = 0

0.02722907	0.07706150	0.12694010	0.17687678	0.22687770	0.27694254	0.32706478	0.37723205	0.42742773	0.47763292
0.52782770	0.57799344	0.62811419	0.67817803	0.72817869	0.77811572	0.82799502	0.87782862	0.92763270	0.97742602

IM = 3 IT1 = 0

0.02718600	0.07699412	0.12685510	0.17678282	0.22678433	0.27685902	0.32699915	0.37719061	0.42741434	0.47764858
0.52787091	0.57805996	0.62819766	0.67827058	0.72827139	0.77819968	0.82806225	0.87787237	0.92764822	0.97741155

11 TIME = 1.7294 MIN.

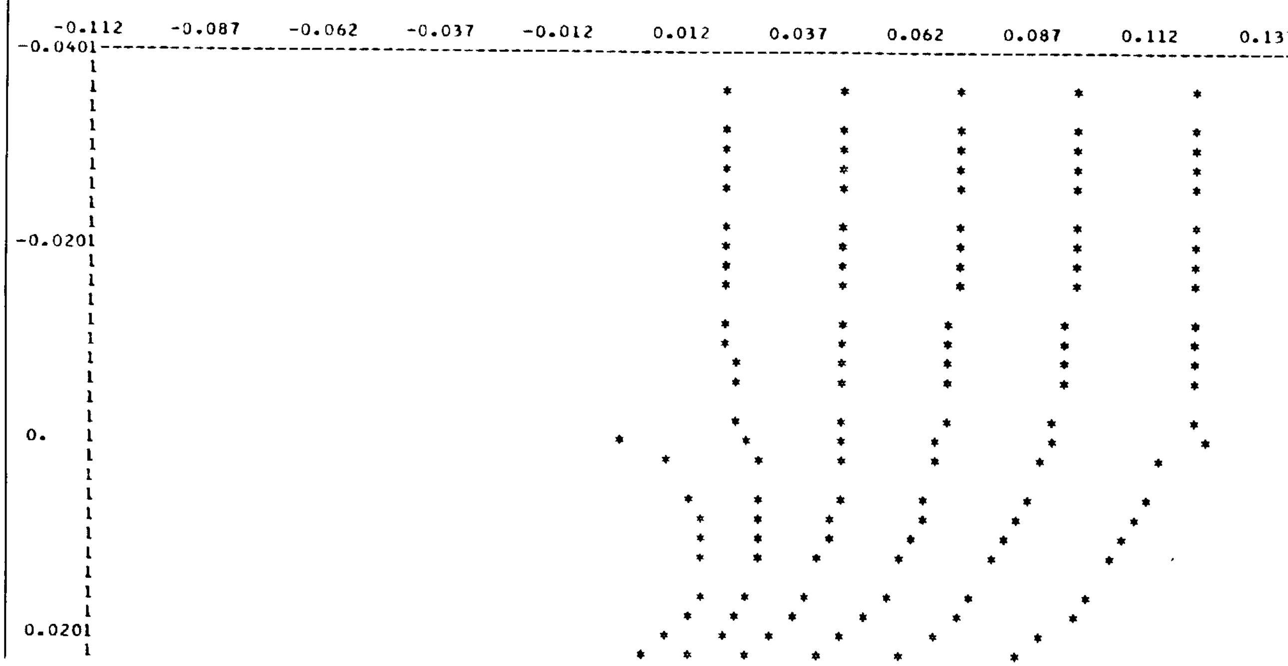
12

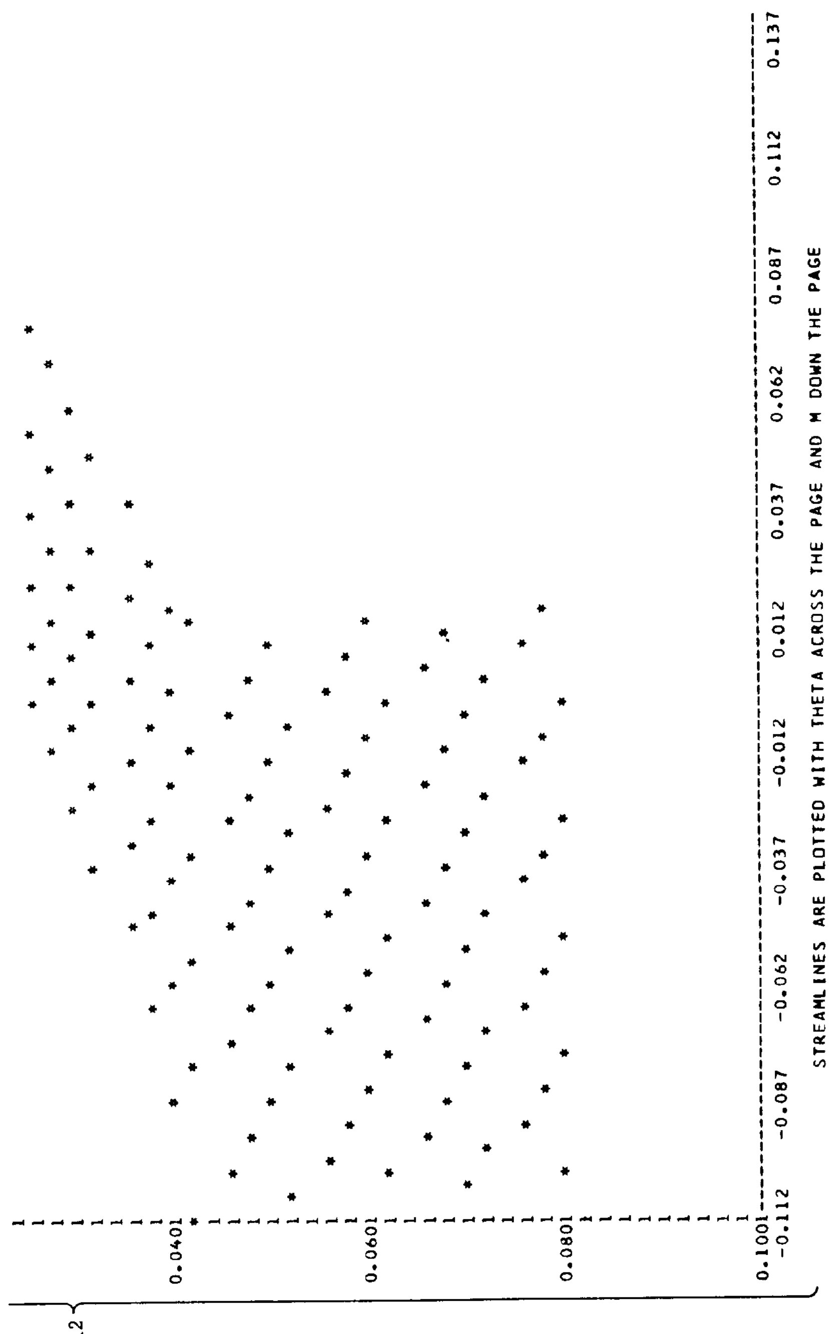
STREAMLINE COORDINATES

M COORDINATE	STREAM FN.	THETA	STREAM FN.	THETA	STREAM FN.	THETA
-0.3512353E-01	0.2000000	0.2175627E-01	0.4000000	0.4683256E-01	0.6000000	0.7187305E-01
	0.8000000	0.9700366E-01	1.0000000	0.1222286	0.2000000	0.2175627E-01
-0.3261471E-01	0.2000000	0.2175627E-01	0.4000000	0.4683255E-01	0.6000000	0.7187300E-01
	0.8000000	0.9700367E-01	1.0000000	0.1222287	0.2000000	0.2175627E-01
-0.3010588E-01	0.2000000	0.2176819E-01	0.4000000	0.4683618E-01	0.6000000	0.7186363E-01
	0.8000000	0.9699390E-01	1.0000000	0.1222321	0.2000000	0.2176819E-01

TABLE I. - Continued. SAMPLE OUTPUT

STREAMLINE PLOTS




STREAMLINES ARE PLOTTED WITH THETA ACROSS THE PAGE AND M DOWN THE PAGE
-0.112
-0.087
-0.062
-0.037
-0.012
0.012
0.037
0.062
0.087
0.112
0.137
0.040
0.060
0.080
0.100
12

VELOCITIES AT INTERIOR MESH POINTS

13

IM= 1

VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)
61.703	-0.01	61.750	-0.02	61.818	-0.03	61.898	-0.03	61.981	-0.03
62.060	-0.03	62.127	-0.02	62.175	-0.01	62.199	-0.00	62.199	0.01
62.173	0.01	62.125	0.02	62.059	0.03	61.981	0.03	61.898	0.03
61.819	0.03	61.751	0.02	61.702	0.01	61.676	0.01	61.678	-0.01

IM= 2

VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)	VELOCITY	ANGLE(DEG)
61.703	0.02	61.750	0.03	61.818	0.04	61.898	0.04	61.981	0.04
62.060	0.04	62.127	0.03	62.175	0.02	62.200	0.01	62.199	-0.01
62.173	-0.02	62.125	-0.03	62.059	-0.04	61.980	-0.04	61.898	-0.04
61.819	-0.04	61.751	-0.03	61.703	-0.02	61.676	-0.01	61.678	0.01

14 ITERATION NO. 1 MAXIMUM RELATIVE CHANGE IN DENSITY = 0.5774

15

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS

		BLADE SURFACE 1					BLADE SURFACE 2				
M	*	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	*	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	*
0	*	0	90.00	0	0	*	0	-90.00	0	0	*
0.2509E-02	*	99.866	27.31	0.4405E-02	0.3215	*	74.794	-19.97	0.4372E-02	0.2408	*
0.5018E-02	*	114.60	21.66	0.7165E-02	0.3689	*	63.478	-13.33	0.6959E-02	0.2043	*
0.7526E-02	*	134.41	14.63	0.9807E-02	0.4327	*	59.630	-15.67	0.9547E-02	0.1920	*
0.1004E-01	*	158.08	6.20	0.1236E-01	0.5089	*	58.400	-20.80	0.1219E-01	0.1880	*

15

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS

BLADE SURFACE 1

M	VELOCITY	ANGLE(DEG)	W/WCR
0	11.357	90.00	0.3656E-01
0.6132E-03	76.923	57.04	0.2476
0.3582E-02	100.30	25.05	0.3229
0.1906E-01	240.79	-29.61	0.7751
0.2201E-01	240.01	-40.10	0.7726

15

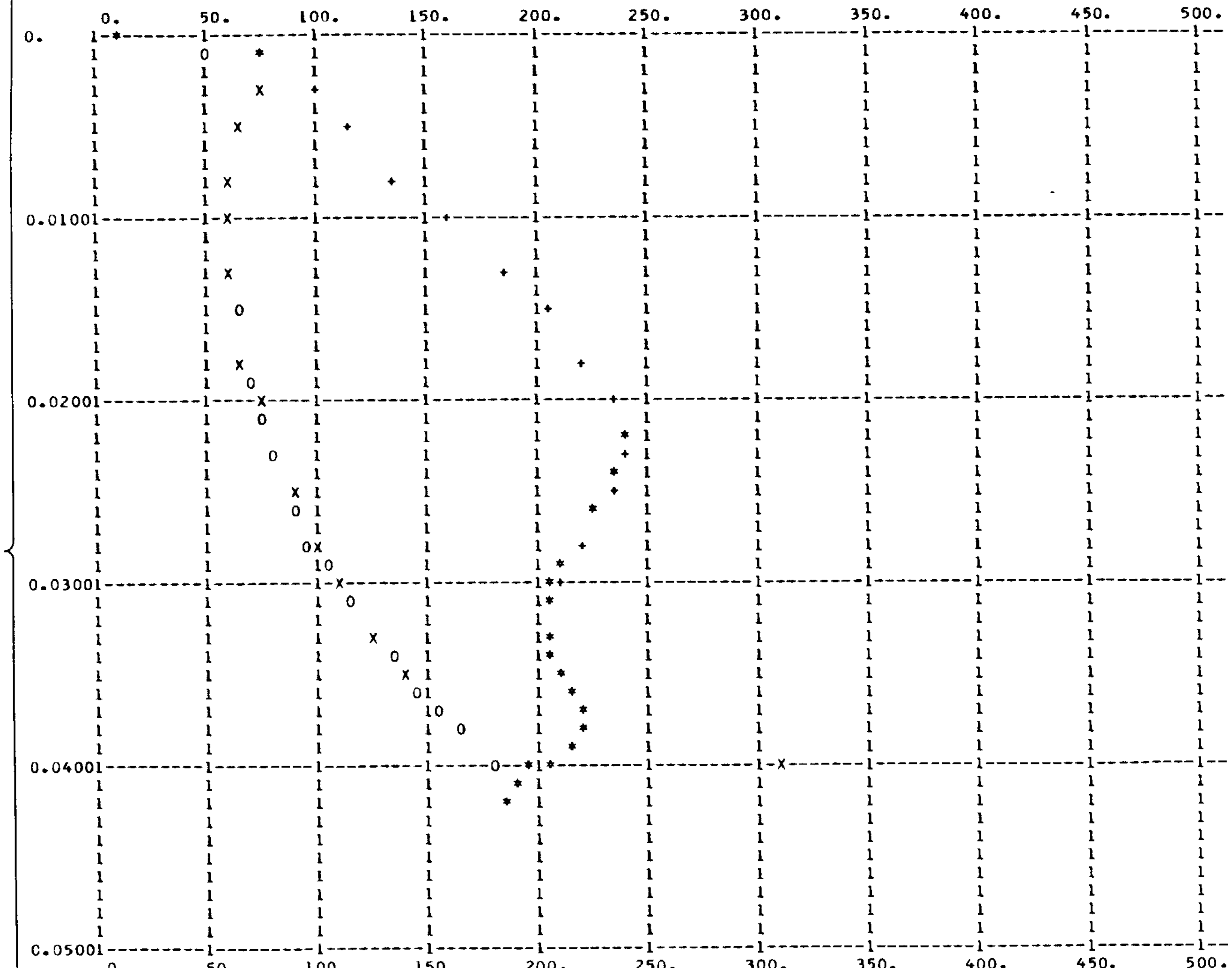
BLADE SURFACE 2

M	VELOCITY	ANGLE(DEG)	W/WCR
0.6132E-03	47.792	-57.04	0.1538
0.4759E-02	48.156	-13.29	0.1550
0.1140E-01	57.949	-23.60	0.1865
0.1539E-01	62.953	-31.07	0.2027

TABLE I. - Concluded. SAMPLE OUTPUT

BLADE SURFACE VELOCITIES

15



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

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

16 TIME = 1.9575 MIN.

Output

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

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

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

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

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

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

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

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

(7) This is the output corresponding to AANDK.

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

(9) This is the output corresponding to ERSOR.

(10) This is the output corresponding to STRFN.

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

(12) This is the output corresponding to SLCRD.

(13) This is the output corresponding to INTVL.

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

(15) This is the output corresponding to SURVL.

(16) This is the total execution time after all calculations are completed for an outer iteration.

ERROR CONDITIONS

(1) SPLINT USED FOR EXTRAPOLATION
EXTRAPOLATED VALUE = X.XXX

SPLINT is normally used for interpolation, but may be used for extrapolation in some cases. When this occurs, the above message is printed as well as the input and output of SPLINT. Calculations proceed normally after this printout.

(2) BLCD CALL NO. XX
M COORDINATE IS NOT WITHIN BLADE

This message is printed by subroutine BLCD if the m-coordinate given this subroutine as input is not within the bounds of the blade surface for which BLCD is called. The value of m and the blade-surface number are also printed when this happens. This may be caused by an error in the integer input items for the program.

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

(3) ROOT CALL NO. XX
ROOT HAS FAILED TO CONVERGE IN 1000 ITERATIONS

This message is printed by subroutine ROOT if a root cannot be located. The input to ROOT is also printed. The user should thoroughly check the input to the main program.

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

(4) DENSTY CALL NO. XX
NER(1) = XX
RHO*W IS X.XXXX TIMES THE MAXIMUM VALUE FOR RHO*W

This message is printed if the value of ρW at some mesh point is so large that there is no solution for the value of ρ and W. This indicates a locally supersonic condition, which can be eliminated by decreasing WTFL in the input.

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

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

(5) MM, NBBI, NRSP, OR SOME SPLNO IS TOO LARGE

If this is printed, reduce the appropriate inputs to their allotted maximum values.

(6) WTFL IS TOO LARGE AT BLADE LEADING EDGE

This is printed if WTFL is greater than the choking mass flow for the boundary BG. If this message is printed, WTFL is cut in half by the program and calculations proceed as usual.

(7) ONE OF THE MH ARRAYS IS TOO LARGE

This is printed if there are more than 100 intersections of horizontal mesh lines with any blade surface. In this case NBBI should be reduced.

(8) THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500

This is printed if there are more than the allowable number of finite-difference grid points. Either MM or NBBI must be reduced.

(9) SEARCH CANNOT FIND M IN THE MH ARRAY.

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

PROGRAM LISTING

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

```
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1   RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      COMMON /BLCDCM/ EM(50,2),INIT(2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      CALL TIME1(T1)
```

```
   10 IEND = -1
      ITER = 0
      INIT(1) = 0
      INIT(2) = 0
      CALL INPUT
      CALL PRECAL
   30 CALL COEF
      CALL SOR
      CALL TIME1(T2)
      TIME= (T2-T1)/3600.
      WRITE(6,1000) TIME
      CALL SLAX
      CALL TANG
      CALL VELOCY
      CALL TIME1(T2)
      TIME= (T2-T1)/3600.
      WRITE(6,1000) TIME
      IF(NER(2).GT.0) GO TO 10
      IF (IEND) 30,30,10
 1000 FORMAT (8HLTIME = ,F7.4,5H MIN.)
      END
```

```
      SUBROUTINE INPUT
C
C   INPUT READS AND PRINTS ALL INPUT DATA CARDS AND CALCULATES HORIZONTAL
C   SPACING (MV ARRAY)
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1   RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C   READ AND PRINT ALL INPUT DATA
C
```

```
      WRITE(6,1000)
      READ(5,1100)
      WRITE(6,1100)
      WRITE(6,1110)
      READ (5,1030) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      WRITE(6,1040) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      WRITE(6,1120)
      READ (5,1030)BETAI,BETAO,CHORD(1),STGR(1)
      WRITE(6,1040)BETAI,BETAO,CHORD(1),STGR(1)
      WRITE(6,1130)
      READ (5,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
      WRITE(6,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP
      DO 10 J=1,2
      IF (J.EQ.1) WRITE(6,1140)
      IF (J.EQ.2) WRITE(6,1150)
      WRITE(6,1180) J,J,J,J,J
      READ (5,1030) RI(J),RO(J),BETI(J),BETO(J),SPLNO
      WRITE(6,1040) RI(J),RO(J),BETI(J),BETO(J),SPLNO
      NSPI(J)= SPLNO
      NSP = NSPI(J)
      WRITE(6,1190) J
      READ (5,1030) (MSP(I,J),I=1,NSP)
      WRITE(6,1040) (MSP(I,J),I=1,NSP)
      WRITE(6,1200) J
      READ (5,1030) (THSP(I,J),I=1,NSP)
   10 WRITE(6,1040) (THSP(I,J),I=1,NSP)
      WRITE(6,1210)
      READ (5,1030) (MR(I),I=1,NRSP)
      WRITE(6,1040) (MR(I),I=1,NRSP)
      WRITE(6,1220)
      READ (5,1030) (RMSP(I),I=1,NRSP)
      WRITE(6,1040) (RMSP(I),I=1,NRSP)
      WRITE(6,1230)
      READ (5,1030) (BESP(I),I=1,NRSP)
      WRITE(6,1040) (BESP(I),I=1,NRSP)
      WRITE(6,1240)
      READ (5,1010) BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      WRITE(6,1020) BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      IF (MM.LE.100.AND.NBBI.LE.50.AND.NRSP.LE.50.AND.NSPI(1).LE.50
     1   .AND.NSPI(2).LE.50) GO TO 20
      WRITE (6,1250)
      STOP
C
C   CALCULATE MV ARRAY
C
   20 HM1 = CHORD(1)/FLOAT(MBO-MBI)
      DO 30 IM=1,MM
   30 MV(IM) = FLOAT(IM-MBI)*HM1
C
C   CALCULATE MISCELLANEOUS CONSTANTS
C
      NER(1)=0
      NER(2)=0
      PITCH = 2.*3.1415927/FLOAT(NBL)
      HT= PITCH/FLOAT(NBBI)
      DTLR= HT/1000.
      DMLR = HM1/1000.
      BV(1) = 0.
      BV(2) = 1.
      MBIM1= MBI-1
      MBIP1= MBI+1
      MBOM1= MBO-1
      MBOP1= MBO+1
      MMM1 = MM-1
```

```
      CP = AR/(GAM-1.)*GAM
      EXPON= 1./(GAM-1.)
      TWW= 2.*OMEGA/WTFL
      CPTIP= 2.*CP*TIP
      TGROG= 2.*GAM*AR/(GAM+1.)
      CALL SPLINT(MR,RMSP,NRSP,MV,MM,RM,SAL)
      CALL SPLINT(MR,BESP,NRSP,MV,MM,BE,DBDM)
C
C  CALCULATE GEOMETRICAL CONSTANTS
C
      CHORD(2) = CHORD(1)
      STGR(2) = STGR(1)
      MLE(1) = 0.
      MLE(2) = 0.
      THLE(1) = 0.
      THLE(2) = PITCH
      RMI(1) = RM(MBI)
      RMI(2) = RM(MBI)
      RMO(1) = RM(MBO)
      RMO(2) = RM(MBO)
C
C  INITIALIZE ARRAYS
C
      DO 60 I=1,2500
      U(I) = 1.
      K(I) = 0.
   60 RHO(I) = RHOIP
      DO 70 IM=1,100
      DO 70 SURF=1,2
      RHOHB(IM,SURF) = RHOIP
   70 RHOVB(IM,SURF) = RHOIP
      RETURN
 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,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRHOIP,12X,4HWTFL,11X,6HW
     1TFLSP,10X,5HOMEGA,12X,3HORF)
 1120 FORMAT (6X,5HBETAI,10X,5HBETAO,11X,6HCHORDF,11X,5HSTGRF)
 1130 FORMAT (41H   MBI  MBO            MM  NBBI  NBL NRSP)
 1140 FORMAT (39HL     BLADE SURFACE 1  --  UPPER SURFACE)
 1150 FORMAT (39HL     BLADE SURFACE 2  --  LOWER SURFACE)
 1180 FORMAT (7X,2HRI,I1,12X,2HRO,I1,12X,4HBETI,I1,11X,4HBETO,I1,11X,5HS
     1PLNO,I1)
 1190 FORMAT (7X,3HMSP,I1,2X,5HARRAY)
 1200 FORMAT (7X,4HTHSP,I1,2X,5HARRAY)
 1210 FORMAT (16HL       MR  ARRAY)
 1220 FORMAT (7X,11HRMSP  ARRAY)
 1230 FORMAT (7X,11HBESP  ARRAY)
 1240 FORMAT (52HL     BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL)
 1250 FORMAT (41H1 MM,NBBI,NRSP,OR SOME SPLNO IS TOO LARGE)
      END
```

```
$IBFTC PRECAL  DEBUG
      SUBROUTINE PRECAL
C
C  PRECAL CALCULATES ALL REQUIRED FIXED CONSTANTS
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      EXTERNAL BL1,BL2
C
C  CALCULATE LAMBDA AND VI
C
      BETAI = BETAI/57.295779
      BETAO = BETAO/57.295779
      TBI = SIN(BETAI)/COS(BETAI)
      TBO = SIN(BETAO)/COS(BETAO)
   10 RHOT = RHOIP
      RHOVI = WTFL/BE(MBI)/PITCH/COS(BETAI)/RM(MBI)
   20 VI = RHOVI/RHOT
      LAMBDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))
      TTIP = 1.-(VI**2+2.*OMEGA*LAMBDA-(OMEGA*RM(MBI))**2)/CPTIP
      IF(TTIP.LE.0.) GO TO 30
      RHOMBI = RHOIP*TTIP**EXPON
      IF(ABS(RHOMBI-RHOT)/RHOIP.LT..000001) GO TO 40
      RHOT = RHOMBI
      GO TO 20
   30 WTFL = WTFL/2.
      NER(2)= NER(2)+1
      WRITE(6,1020) WTFL
      IF(NER(2).EQ.10) STOP
      GO TO 10
   40 VI = RHOVI/RHOMBI
      LAMBDA = RM(MBI)*(VI*SIN(BETAI)+OMEGA*RM(MBI))
C
C  CALCULATE MAXIMUM VALUES FOR RHO*W AT LEADING AND TRAILING EDGE
C
      TWL = 2.*OMEGA*LAMBDA
      AA = (TWL-(OMEGA*RM(MBI))**2)/CPTIP
      TPP = TIP*(1.-AA)
      BB = TGROG*TPP
      TTIP = 1.-BB/CPTIP-AA
      RHOT = RHOIP*TTIP**EXPON
      RHOWMI = RHOT*SQRT(BB)
      AA = (TWL-(OMEGA*RM(MBO ))**2)/CPTIP
      TPP = TIP*(1.-AA)
      BB = TGROG*TPP
      TTIP = 1.-BB/CPTIP-AA
      RHOT = RHOIP*TTIP**EXPON
      RHOWMO = RHOT*SQRT(BB)

   CALCULATE VO AND W-CRITICAL AT BLADE LEADING AND TRAILING EDGE
```

```
      RHOVO = WTFL/BE(MBO )/PITCH/COS(BETAO)/RM(MBO)
      RHOMB2 = RHOIP
      TWLMR = TWL-(OMEGA*RM(MBO ))**2
      LER(1)=1
C     DENSTY CALL NO. 1
      CALL DENSTY(RHOVO,RHOMB2,VO,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      WCRI = SQRT(TGROG*TIP*(1.-(TWL-(OMEGA*RM(MBI))**2)/CPTIP))
      WCRO = SQRT(TGROG*TIP*(1.-(TWL-(OMEGA*RM(MBO ))**2)/CPTIP))
C
C    CALCULATE BETA CORRECTED TO BOUNDARY A-N AND G-H
C
      TWLMR = TWL-(OMEGA*RM(1))**2
      RHO1 = RHOMBI
      TBI1 = 1.E20
   50 TBIT = (TBI/BE(MBI)*RHO1/RHOMBI+OMEGA*(RM(MBI)**2-RM(1)**2)*RHO1
     1   /WTFL*PITCH)*BE(1)
      IF(ABS(TBI1-TBIT).LT..00001) GO TO 60
      TBI1 = TBIT
      RHOVI = WTFL/PITCH*SQRT(1.+TBI1**2)/BE(1)/RM(1)
      LER(1)=2
C     DENSTY CALL NO. 2
      CALL DENSTY (RHOVI,RHO1,AA,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      GO TO 50
   60 TBI = TBIT
      BTAIN = ATAN(TBI)*57.295779
      TWLMR = TWL-(OMEGA*RM(MM))**2
      RHOMM = RHOMB2
      TBOM = 1.E20
   70 TBOT = (TBO/BE(MBO )*RHOMM/RHOMB2+OMEGA*(RM(MBO )**2-RM(MM)**2)*
     1   RHOMM/WTFL*PITCH)*BE(MM)
      IF (ABS(TBOM-TBOT).LT..00001) GO TO 80
      TBOM = TBOT
      RHOVO = WTFL/PITCH*SQRT(1.+TBOM**2)/BE(MM)/RM(MM)
      LER(1)=3
C     DENSTY CALL NO. 3
      CALL DENSTY (RHOVO,RHOMM,AA,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
      GO TO 70
   80 TBO = TBOT
      BTAOUT = ATAN(TBO)*57.295779
C
C  CALCULATE TV, ITV, IV, DTDMV, AND BETAV ARRAYS
C
    . ITMIN = 0
      ITMAX = NBBI-1
C  TV, ITV, AND DTDMV ON BLADE
      DO 90  IM=MBI,MBO
      LER(2)=1
C     BLCD CALL NO. 1
      CALL BL1(MV(IM),TV(IM,1),DTDMV(IM,1),INF)
      ITV(IM,1)= INT((TV(IM,1)+DTLR)/HT)
      IF (TV(IM,1).GT.-DTLR) ITV(IM,1)=ITV(IM,1)+1
      ITMIN= MINO(ITMIN,ITV(IM,1))
      LER(2)=2
C     BLCD CALL NO. 2
      CALL BL2(MV(IM),TV(IM,2),DTDMV(IM,2),INF)
      ITV(IM,2)= INT((TV(IM,2)-DTLR)/HT)
      IF (TV(IM,2).LT.DTLR) ITV(IM,2)=ITV(IM,2)-1
   90 ITMAX= MAXO(ITMAX,ITV(IM,2))
C    ITV AND IV UPSTREAM OF BLADE
      FIRST = 0
      LAST = NBBI-1
      DO 120 IM=1,MBIM1
      ITV(IM,1)= FIRST
  120 ITV(IM,2)= LAST
C    ITV DOWNSTREAM OF BLADE
```

```
  140 LAST= ITV(MBO,2)
      FIRST= LAST+1-NBBI
      DO 150 IM=MBOP1,MM
      ITV(IM,1) = FIRST
  150 ITV(IM,2) = LAST
      ITMIN = MINO(ITMIN,ITV(MM,1))
C   CALCULATE IV ARRAY
      IV(1) = 1
      DO 160 IM=1,MM
  160 IV(IM+1) = IV(IM)+ITV(IM,2)-ITV(IM,1)+1
C  BETAV ARRAY
      DO 200 SURF=1,2
      DO 200 IM=MBI,MBO
  200 BETAV(IM,SURF) = ATAN(DTDMV(IM,SURF)*RM(IM))*57.295779
      NIP = IV(MM)+NBBI-1
      WRITE(6,1030) VI,RHOWMI,WCRI,BTAIN,VO,RHOWMO,WCRO,BTAOUT
      WRITE(6,1040) PITCH,HT,HM1
      WRITE(6,1050) ITMIN,ITMAX,LAMBDA,NIP
      WRITE(6,1060) (SURF,BV(SURF),SURF=1,2)
      IF(BLDAT.LE.0) GO TO 230
      WRITE (6,1070)
      WRITE (6,1080) (MV(IM),TV(IM,1),DTDMV(IM,1),TV(IM,2),DTDMV(IM,2),
     1   IM=MBI,MBO)
      WRITE (6,1090) (IM,MV(IM),RM(IM),SAL(IM),BE(IM),DBDM(IM),IM=1,MM)
  230 CONTINUE
C
C  CALCULATE MH AND DTDMH ARRAYS
C
      ITO = ITV(1,1)
      MRTS = 1
      IMS(1) = 1
      MH(1,1) = 0.
      DTDMH(1,1) = 1.E10
      LER(2) = 3
C     BLCD AND ROOT (VIA MHORIZ) CALL NO. 3
      CALL MHORIZ(MV,ITV(1,1),BL1,MBI,MBO,ITO,HT,DTLR,0,IMS(1),MH(1,1),
     1   DTDMH(1,1),MRTS)
      IF (ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) GO TO 240
      IMSL = IMS(1)+1
      MH(IMSL,1) = MV(MBO)
      DTDMH(IMSL,1) = -1.E10
      IMS(1) = IMSL
  240 IMS(2) = 0
      MRTS = 1
      LER(2) = 4
C     BLCD AND ROOT (VIA MHORIZ) CALL NO. 4
      CALL MHORIZ(MV,ITV(1,2),BL2,MBI,MBO,ITO,HT,DTLR,1,IMS(2),MH(1,2),
     1   DTDMH(1,2),MRTS)
      I = MAXO(IMS(1),IMS(2))
      IF(I.LE.100) GO TO 290
      WRITE(6,1100) I
      STOP
  290 IF(BLDAT.LE.0) GO TO 300
      WRITE (6,1110)  (IM,IV(IM),(ITV(IM,SURF),SURF=1,2),IM=1,MM)
C
C  CALCULATE RMH, BEH, AND BETAH ARRAYS
C
  300 IF(BLDAT.GT.0) WRITE(6,1120)
      DO 320 SURF=1,2
      CALL SPLINT(MR,RMSP,NRSP,MH(1,SURF),IMS(SURF),RMH(1,SURF),AAA)
      CALL SPLINT(MR,BESP,NRSP,MH(1,SURF),IMS(SURF),BEH(1,SURF),AAA)
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 320
      DO 310 IHS = 1,IMSS
```

```
 310 BETAH(IHS,SURF) = ATAN(DTDMH(IHS,SURF)*RMH(IHS,SURF))*57.295779
     IF (BLDAT.GT.0) WRITE(6,1130) SURF,(MH(IM,SURF),RMH(IM,SURF),
    1   BEH(IM,SURF),BETAH(IM,SURF),DTDMH(IM,SURF),IM=1,IMSS)
 320 CONTINUE
     IF (BLDAT.LE.0) GO TO 340
     WRITE (6,1140)
     IT = ITMIN
 330 IF (IT.GT.ITMAX) GO TO 340
     TH = FLOAT(IT)*HT
     WRITE (6,1010) IT,TH
     IT = IT+1
     GO TO 330
 340 IF(NIP.LE.2500) GO TO 350
     WRITE(6,1150)
     STOP
 350 WRITE (6,1000)
     RETURN
1000 FORMAT (1H1)
1010 FORMAT (4X,I4,G16.5)
1020 FORMAT(60HLINPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE LEADING
    1EDGE/16H WTFL REDUCED TO,G14.6)
1030 FORMAT (1H1/24X,10HFREESTREAM,8X,13HMAXIMUM VALUE,
    17X,8HCRITICAL,30X,14HBETA CORRECTED/25X,8HVELOCITY,10X,9HFOR RHO*W
    2,10X,8HVELOCITY,31X,11HTO BOUNDARY/1X,17HLEADING EDGE  B-G,3G18.5,
    312X,12HBOUNDARY A-H,G18.5/1X,17HTRAILING EDGE C-F,3G18.5,12X,
    412HBOUNDARY D-E,G18.5)
1040 FORMAT(33HL     CALCULATED PROGRAM CONSTANTS//5X,5HPITCH,13X,
    1   2HHT,13X,3HHM1/1X,5G16.7)
1050 FORMAT (/5X,5HITMIN,10X,5HITMAX/4X,I5,10X,I5//5X,6HLAMBDA/1X,G16.7
    1   ,/38HL     NUMBER OF INTERIOR MESH POINTS = ,I5)
1060 FORMAT(28HL     SURFACE BOUNDARY VALUES//5X,7HSURFACE,7X,2HBV
    1/(5X,I4,4X,F10.5))
1070 FORMAT (1H1,6X,62HBLADE DATA AT INTERSECTIONS OF VERTICAL MESH LIN
    1ES WITH BLADES)
1080 FORMAT (1HL,22X,15HBLADE SURFACE 1,15X,15HBLADE SURFACE 2/7X,
    1   1HM,14X,2HTV,11X,5HDTDMV,12X,2HTV,11X,5HDTDMV/(5G15.5))
1090 FORMAT (1H1,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABLE /
    1   2X,2HIM,7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HDB/DM/(1X,I3,
    2   5G15.5))
1100 FORMAT(34HLONE OF THE MH ARRAYS IS TOO LARGE/7H IT HAS,I5, 8H  POI
    1NTS)
1110 FORMAT (4H1 IM,9X,8HIV ARRAY,25X,9HITV ARRAY/38X,5HBLADE/37X,7HSUR
    1FACE,3X,1H1,5X,1H2/39X,3HNO./(1X,I3,5X,I10,25X,2(I4,2X)))
1120 FORMAT (67H1M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINE
    1S WITH BLADE)
1130 FORMAT (25HLMH ARRAY - BLADE SURFACE,I2//15X,2HMH,19X,3HRMH,19X,
    1   3HBEH,18X,5HBETAH,17X,5HDTDMH/(5G22.4))
1140 FORMAT (43H1THETA COORDINATES OF HORIZONTAL MESH LINES//6X,2HIT,
    15X,5HTHETA)
1150 FORMAT(48HLTHE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500)
     END
```

```
$IBFTC MHORIZ  DEBUG
      SUBROUTINE MHORIZ(MV,ITV,BL,MBI,MBO,ITO,HT,DTLR,KODE,J,MH,DTDMH,
     1MRTS)
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(2)
      DIMENSION MV(100),ITV(100),MH(100),DTDMH(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      REAL MVIM
      EXTERNAL BL
      IF (MBI.GE.MBO) RETURN
      IM= MBI
   10 ITIND= 0
   20 IF (ITV(IM+1)-ITV(IM)-ITIND) 30,40,50
   30 J= J+1
      TI= FLOAT(ITV(IM+1)-ITO-ITIND+KODE)*HT
      ITIND= ITIND-1
      MVIM = MV(IM)
      IF (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
```

```
$IBFTC COEF    DEBUG
      SUBROUTINE COEF
C
C  COEF CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  AT ALL UNKNOWN MESH POINTS FOR THE ENTIRE REGION
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C  INITIALIZE ARRAYS
      ITER = ITER+1
      IH(1) = 1
      IH(2) = 0
C  INCOMPRESSIBLE CASE
      IF(GAM.NE.1.5.OR.AR.NE.1000..OR.TIP.NE.1.E6) GO TO 20
      IEND = 1
      GO TO 40
C  ADJUSTMENT OF PRINTING CONTROL VARIABLES
   20 IF(ITER.NE.1.AND.ITER.NE.2) GO TO 30
      AANDK = AANDK-1
      ERSOR = ERSOR-1
      STRFN = STRFN-1
      SLCRD = SLCRD-1
      INTVL = INTVL-1
      SURVL = SURVL-1
   30 IF(IEND.NE.0) GO TO 40
      AANDK = AANDK+2
      ERSOR = ERSOR+2
      STRFN = STRFN+2
      SLCRD = SLCRD+2
      INTVL = INTVL+2
      SURVL = SURVL+2
C
C  FIRST VERTICAL MESH LINE
C
   40 DO 50 IP=1,NBBI
      A(IP,1) = 0.
      A(IP,2) = 0.
      A(IP,3) = 0.
```

```
      A(IP,4) = 1.
   50 K(IP) = HM1*TBI/PITCH/RM(1)
C
C   UPSTREAM OF BLADE, EXCEPT FOR FIRST VERTICAL MESH LINE
C
      IF(2.GT.MBIM1) GO TO 70
      DO 60 IM=2,MBIM1
   60 CALL COEFP(IM)
C
C   BETWEEN BLADES
C
   70 DO 80 IM=MBI,MBO
   80 CALL COEFBB(IM)
C
C  DOWNSTREAM OF BLADES EXCEPT FOR FINAL MESH LINE
C
  150 IF(MBOP1.GT.MMM1) GO TO 170
      DO 160 IM=MBOP1,MMM1
  160 CALL COEFP(IM)
C
C  FINAL VERTICAL MESH LINE
C
  170 IVMM = IV(MM)
      DO 180 IP=IVMM,NIP
      A(IP,1) = 0.
      A(IP,2) = 0.
      A(IP,3) = 1.
      A(IP,4) = 0.
  180 K(IP) = -HM1*TBO/PITCH/RM(MM)
C
C  TAKE CARE OF POINTS ADJACENT TO B, AND CASES WHEN POINTS J,C,E, OR F
C  ARE GRID POINTS
C
C  POINT B
      IP = IV(MBIM1)
      A(IP,4) = 0.
C  POINT C
      IF(ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) RETURN
      IT = ITV(MBO,1)-1
      IP = IPF(MBOP1,IT)
      A(IP,3) = 0.
      RETURN
      END
```

```
$IBFTC COEFBB   DEBUG
      SUBROUTINE COEFBB(IM)
C
C   COEFBB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K
C   ALONG ALL VERTICAL MESH LINES WHICH INTERSECT BLADES
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      IF(ITV(IM,1).GT.ITV(IM,2)) RETURN
      ITVU = ITV(IM,1)
      IT   = ITVU - 1
      ITVL = ITV(IM,2)
      IPU = IPF(IM,ITVU)
      IPL = IPU+ITVL-ITVU
      DO 90 IP=IPU,IPL
      IT = IT+1
      CALL HRB(IM,IT,IP)
      DO 10 I=1,4
      KAK(I) = 0.
   10 KA(I) = 0
C   FIX HRB VALUES FOR CASES WHERE MESH LINES INTERSECT BLADES
   60 IF(IT.EQ.ITV(IM,1)) CALL BDRY12(1,IM,IT)
      IF(IT.EQ.ITV(IM,2)) CALL BDRY12(2,IM,IT)
      ITVM1 = ITV(IM-1,1)
      ITVP1 = ITV(IM+1,1)
      IF(IT.LT.ITVM1) CALL BDRY34(3,IM,1)
      IF(IT.LT.ITVP1) CALL BDRY34(4,IM,1)
      IF(IT.GT.ITV(IM-1,2)) CALL BDRY34(3,IM,2)
      IF(IT.GT.ITV(IM+1,2)) CALL BDRY34(4,IM,2)
   70 IF(IM.EQ.MBO.AND.LOWER.EQ.2) GO TO 80
C   COMPUTE A AND K COEFFICIENTS
   80 CALL AAK(IM,IP)
      DO 90 I=1,4
      K(IP) = K(IP)+KAK(I)*A(IP,I)
   90 IF(KA(I).EQ.1) A(IP,I) = 0.
      RETURN
C
C   COEFP CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C   ALONG ALL VERTICAL MESH LINES WHICH DO NOT INTERSECT BLADES
```

```
C
      ENTRY COEFP(IM)
      ITVU = ITV(IM,1)
      IT = ITVU-1
      ITVL = ITV(IM,2)
      IPL = IV(IM+1)-1
      IPU = IV(IM)
      DO 100 IP=IPU,IPL
      IT = IT+1
      CALL HRB(IM,IT,IP)
      IF (IT.EQ.ITVU) R(1) = RHO(IPL)
      IF (IT.EQ.ITVL) R(2) = RHO(IPU)
  100 CALL AAK(IM,IP)
      K(IPL) = K(IPL)+A(IPL,2)
      K(IPU) = K(IPU)-A(IPU,1)
      RETURN
      END
```

```
$IBFTC HRB      DEBUG
      SUBROUTINE HRB(IM,IT,IP)
C
C  HRB CALCULATES MESH SPACING, H, DENSITIES, RZ AND R, AT GIVEN AND
C  ADJACENT POINTS, AND STREAM SHEET THICKNESSES, BZ AND B, AT GIVEN
C  AND ADJACENT POINTS
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,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)
      R(2) = RHO(IP+1)
      R(3) = RHO(IP3)
      R(4) = RHO(IP4)
      BZ= BE(IM)
      B(3)= BE(IM-1)
      B(4)= BE(IM+1)
      RETURN
      END
```

```
$IBFTC AAK      DEBUG

      SUBROUTINE AAK(IM,IP)
C
C  AAK CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANT, K,
C  AT A SINGLE MESH POINT
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      A12= 2./H(1)/H(2)
      A34= 2./H(3)/H(4)
      AZ= A12+A34
      B12= (R(2)-R(1))/RZ/(H(1)+H(2))
      B34= (B(4)*R(4)-B(3)*R(3))/BZ/RZ/(H(3)+H(4))-SAL(IM)/RM(IM)
      A(IP,1) = (2./H(1)+B12)/AZ/(H(1)+H(2))
      A(IP,2) = A12/AZ-A(IP,1)
      A(IP,3) = (2./H(3)+B34)/AZ/(H(3)+H(4))
      A(IP,4) = A34/AZ-A(IP,3)
      K(IP) = -TWW*BZ*RZ*SAL(IM)/AZ
      RETURN
      END
```

```
$IBFTC BDRY12   DEBUG

      SUBROUTINE BDRY12(I,IM,IT)
C
C  BDRY12 CORRECTS VALUES COMPUTED BY HRB WHEN A VERTICAL MESH LINE
C  INTERSECTS A BLADE
C
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      H(I) = ABS(FLOAT(IT)*HT-TV(IM,I))*RM(IM)
      R(I)= RHOVB(IM,I)
      KAK(I) = BV(I)
      KA(I)=1
      RETURN
      END
```

```
$IBFTC BDRY34  DEBUG

      SUBROUTINE BDRY34(I,IM,SURF)
C
C  BDRY34 CORRECTS VALUES COMPUTED BY HRB WHEN A HORIZONTAL MESH LINE
C  INTERSECTS A BLADE
C
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      IH(SURF)=IH(SURF)+1
      IHS=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
```

```
$IBFTC SOR     DEBUG

      SUBROUTINE SOR
C
C  SOR SOLVES THE SET OF SIMULTANEOUS EQUATIONS FOR THE STREAM FUNCTION
C  USING THE METHOD OF SUCCESSIVE OVER-RELAXATION
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      AATEMP = AANDK
      IF (ORF.GE.2.) ORF=0.
      IF(ORF.GT.1.) GO TO 50
      ORF = 1.
      ORFOPT= 2.
   40 ORFTEM=ORFOPT
      LMAX = 0.
   50 IF(AATEMP.GT.0) WRITE(6,1010)
      ERROR = 0.
C
C  SOLVE MATRIX EQUATION BY SOR, OR CALCULATE OPTIMUM OVERRELAXATION
C  FACTOR
C
```

```
      IP = 0
      DO 120 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      IT = ITV(IM,1)
      IF(AATEMP.GT.0) WRITE (6,1020) IM,IT
      DO 120 IP=IPU,IPL
      IP1 = IP-1
      IP2 = IP+1
C  CORRECT IP1 AND IP2 ALONG PERIODIC BOUNDARIES
      IF(IM.GE.MBI.AND.IM.LE.MBO) GO TO 60
      IF(IT.EQ.ITV(IM,1)) IP1 = IP1+NBBI
      IF(IT.EQ.ITV(IM,2)) IP2 = IP2-NBBI
   60 IT3 = IT
      IT4 = IT
  100 IP3 = IPF(IM-1,IT3)
      IP4 = IPF(IM+1,IT4)
      IF(ORF.GT.1.) GO TO 110
C  CALCULATE NEW ESTIMATE FOR LMAX
      UNEW = A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4)
      IF (UNEW.LT.1.E-25) U(IP) = 0.
      IF (U(IP).EQ.0.) GO TO 115
      RATIO = UNEW/U(IP)
      LMAX= AMAX1(RATIO,LMAX)
      U(IP) = UNEW
      GO TO 115
C  CALCULATE NEW ESTIMATE FOR STREAM FUNCTION BY SOR
  110 CHANGE = ORF*(K(IP)-U(IP)+A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*
     1   U(IP3)+A(IP,4)*U(IP4))
      ERROR= AMAX1(ERROR,ABS(CHANGE))
      U(IP) = U(IP)+CHANGE
  115 IF(AATEMP.LE.0) GO TO 120
      WRITE (6,1030) IT,IP,IP1,IP2,IP3,IP4,(A(IP,I),I=1,4),K(IP)
  120 IT = IT+1
      AATEMP = 0
      IF(ORF.GT.1.) GO TO 130
      ORFOPT= 2./(1.+SQRT(ABS(1.-LMAX)))
      WRITE(6,1000) ORFOPT
      IF(ORFTEM-ORFOPT.GT..00001.OR.ORFOPT.GT.1.999) GO TO 40
      WRITE (6,1070)
      ORF = ORFOPT
      GO TO 50
  130 IF(ERSOR.GT.0) WRITE(6,1040) ERROR
      IF(ERROR.GT..000001) GO TO 50
      IF(STRFN.LE.0) RETURN
C
C  PRINT STREAM FUNCTION VALUES FOR THIS ITERATION
C
      WRITE (6,1050)
      DO 140 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      ITVU = ITV(IM,1)
      WRITE (6,1020) IM,ITVU
  140 WRITE (6,1060) (U(IP),IP=IPU,IPL)
      RETURN
 1000 FORMAT(24H ESTIMATED OPTIMUM ORF =,F9.6)
 1010 FORMAT (82H1  IT    IP    IP1    IP2    IP3    IP4     A(1)       A(2)
     1    A(3)       A(4)        K)
 1020 FORMAT(5HKIM =,I4,6X,6HIT1 = ,I4)
 1030 FORMAT(1X,I4,5I6,5F10.5)
 1040 FORMAT(8H ERROR =,F11.8)
 1050 FORMAT(1H1,10X,22HSTREAM FUNCTION VALUES)
 1060 FORMAT (2X,10F13.8)
 1070 FORMAT (1H1)
      END
```

```
$IBFTC SLAX    DEBUG
      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 /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /SLA/TSL(600),UINT(6)
      DIMENSION MSL(100),KKK(14),P(4)
      DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
     1   XDOWN(400),YACROS(400)
      EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
     1   (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
     2   (K(1),YACROS(1))
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA (KKK(J),J=4,14,2)/6*1H*/
C
C  CALL SLAVP AND SLAVBB THROUGHOUT THE REGION
C
      ITVU= ITV(1,1)
      ITVL= ITV(1,2)
      DO 10 IM=1,MBIM1
   10 CALL SLAVP(IM,ITVU,ITVL)
      DO 20 IM=MBI,MBO
      I= 0
   20 CALL SLAVBB(IM)
   90 ITVU = ITV(MBOP1,1)
      ITVL = ITV(MBOP1,2)
      DO 100 IM=MBOP1,MM
  100 CALL SLAVP(IM,ITVU,ITVL)
C
C  PLOT STREAMLINES
C
      IF (SLCRD.LE.0) RETURN
      DO 110 IM=1,MM
  110 MSL(IM) = MV(IM)
      KKK(1) = 7
      KKK(2) = 6
      KKK(3) = MM
      P(1) = 1.
      P(3) = 0.
      P(4) = 0.
      WRITE(6,1000)
      CALL PLOTMY(MSL,TSL,KKK,P)
      WRITE(6,1010)
      RETURN
 1000 FORMAT (2HPT,50X,16HSTREAMLINE PLOTS   )
 1010 FORMAT (2HPL,40X,70HSTREAMLINES ARE PLOTTED WITH THETA ACROSS THE
     1PAGE AND M DOWN THE PAGE)
      END
```

```
$IBFTC SLAV    DEBUG
      SUBROUTINE SLAV
C
C  SLAV CALCULATES RHO*W-SUB-M THROUGHOUT THE REGION AND ON THE BLADE
C  SURFACES, AND CALCULATES THE STREAMLINE LOCATIONS
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,N2L,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /SLA/TSL(600),UINT(6)
      DIMENSION TSP(50),USP(50),DUDT(50),TINT(6)
      DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
     1   XDOWN(400),YACROS(400)
      EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
     1   (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
     2   (K(1),YACROS(1))
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  SLAVP CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT
C  INTERSECT BLADES
C
      ENTRY SLAVP(IM,ITVU,ITVL)
      LOC= 0
      NSP= ITVL-ITVU+2
      IP = IV(IM)-1
      DO 10 IT=1,NSP
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   10 USP(IT)= U(IP)
      USP(NSP) = USP(1)+1.
      IP = IV(IM)
      INTU = INT(U(IP)*5.)
      IF (U(IP).GT.0.) INTU=INTU+1
      DO 20 J=1,5
      UINT(J) = FLOAT(INTU)/5.
   20 INTU = INTU+1
      UINT(6) = UINT(1)
      GO TO 100
C
C  SLAVBB CALCULATES ALONG VERTICAL MESHLINES WHICH INTERSECT BLADES
C
```

```
      ENTRY SLAVBB(IM)
      LOC= 1
      ITVUP1 = ITV(IM,1)
      ITVLM1 = ITV(IM,2)
      ITVU = ITVUP1-1
      ITVL = ITVLM1+1
      NSP = ITVL-ITVU+1
      TSP(1) = TV(IM,1)
      TSP(NSP) = TV(IM,2)
      USP(1) = BV(1)
      USP(NSP) = BV(2)
      IP = IV(IM)-1
      NSPM1 = NSP-1
      IF(2.GT.NSPM1) GO TO 70
      DO 60 IT=2,NSPM1
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   60 USP(IT) = U(IP)
   70 DO 80 I=1,6
   80 UINT(I) = FLOAT(I-1)/5.
C
C  FOR BOTH SLAVP AND SLAVBB, CALCULATE RHO*W-SUB-M IN THE REGION, AND
C  RHO*W AT VERTICAL MESH LINE INTERSECTIONS ON THE BLADE SURFACES
C
  100 CALL SPLINE(TSP,USP,NSP,DUDT,AAA)
      IT = LOC
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      DO 110 IP=IPU,IPL
      IT = IT+1
  110 RWM(IP)  = DUDT(IT)*WTFL/BE(IM)/RM(IM)
  120 IF (LOC.EQ.0) GO TO 130
      WMB(IM,1) = DUDT(   1)*WTFL/BE(IM)/RM(IM)
      WMB(IM,2) = DUDT(NSP)*WTFL/BE(IM)/RM(IM)
      RMDTU2 = (RM(IM)*DTDMV(IM,1))**2
      RMDTL2 = (RM(IM)*DTDMV(IM,2))**2
      IF (RMDTU2.GT.10000.) WMB(IM,1) = 0.
      IF (RMDTL2.GT.10000.) WMB(IM,2) = 0.
      WMB(IM,1) = ABS(WMB(IM,1))*SQRT(1.+RMDTU2)
      WMB(IM,2) = ABS(WMB(IM,2))*SQRT(1.+RMDTL2)
  130 IF (SLCRD.LE.0) RETURN
      NI = 6
      CALL SPLINT(USP,TSP,NSP,UINT,NI,TINT,AAA)
      DO 140 J=1,6
      L= (J-1)*MM+IM
  140 TSL(L)= TINT(J)
      IF (IM.EQ.1) WRITE(6,1000)
      WRITE(6,1010) MV(IM),(UINT(J),TINT(J),J=1,6)
      RETURN
 1000 FORMAT(1H1/30X,22HSTREAMLINE COORDINATES/17HL     M COORDINATE,
     1    3(7X,10HSTREAM FN.,10X,5HTHETA,4X)//)
 1010 FORMAT(1X,7G18.7/(19X,6G18.7))
      END
```

```
$IBFTC TANG     DECK
      SUBROUTINE TANG
C
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,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      DIMENSION SPM(100),USP(100),DUDM(100)
      DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
     1   XDOWN(400),YACROS(400)
      EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
     1   (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
     2   (K(1),YACROS(1))
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      EXTERNAL BL1,BL2
C
C  PERFORM CALCULATIONS ALONG ONE HORIZONTAL LINE AT A TIME
C
      IT = ITMIN
   10 IF (IT.GT.ITMAX) RETURN
      S1 = 0
C
C  ON THE GIVEN HORIZONTAL MESH LINE, FIND A FIRST POINT IN THE REGION
C
      IF(IT.GE.0.AND.IT.LT.NBBI) GO TO 60
      IM = MBIM1
   20 IM= IM+1
      IF(IM.GT.MBO) GO TO 200
      SURF = 1
      IF(IT.GE.ITV(IM,1).AND.IT.LT.ITV(IM-1,1)) GO TO 70
      IF(IM.EQ.MBOP1.AND.IT.EQ.ITV(MBO,1)-1.AND.ITV(MBO,1)-ITV(MBO,2)
     1   +NBBI.EQ.2) GO TO 70
      SURF = 2
      IF(IT.LE.ITV(IM,2).AND.IT.GT.ITV(IM-1,2)) GO TO 70
      GO TO 20
C
C  FIRST POINT IS ON BOUNDARY A-H
C
   60 IM1= 1
      IM = 1
      SPM(1) = MV(1)
      USP (1) = U(IT+1)
```

```
      GO TO 90
C
C  FIRST POINT IS ON A BLADE SURFACE
C
   70 S1 = SURF
      IM1 = IM-1
      IM2 = IM
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IM1)
      IF (IM.EQ.MBIP1) MVIM1 = MVIM1+(MV(IM2)-MVIM1)/1000.
      LER(2) = 5
C     BLCD (VIA ROOT) CALL NO. 5
      IF(S1.EQ.1.AND.IM1.NE.MBO)CALL ROOT(MVIM1,MV(IM2),TH,BL1,DTLR,
     1   ANS,AAA)
      LER(2) = 6
C     BLCD (VIA ROOT) CALL NO. 6
      IF(S1.EQ.2)CALL ROOT(MVIM1,MV(IM2),TH,BL2,DTLR,ANS,AAA)
      IF(S1.EQ.1.AND.IM1.EQ.MBO) ANS = MV(MBO)
      SPM(IM1) = ANS
      USP(IM1)= BV(S1)
C
C  MOVE ALONG HORIZONTAL MESH LINE UNTIL MESH LINE INTERSECTS BOUNDARY
C
   90 IF(IM.LT.MBI.OR.IM.GT.MBO) GO TO 120
      SURF = 1
      IF(IT.LT.ITV(IM,SURF).AND.IT.GE.ITV(IM-1,SURF)) GO TO 140
      SURF = 2
      IF(IT.GT.ITV(IM,SURF).AND.IT.LE.ITV(IM-1,SURF)) GO TO 140
  120 SPM(IM) = MV(IM)
      IP = IPF(IM,IT)
      USP(IM) = U(IP)
      IF (IM.EQ.MM) GO TO 130
      IM= IM+1
      GO TO 90
C
C   FINAL POINT IS ON BOUNDARY D-E
C
  130 IMT = MM
      GO TO 150
C
C  FINAL POINT IS ON A BLADE SURFACE
C
  140 ST = SURF
      IMT=IM
      IMTM1= IMT-1
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IMTM1)
      IF (IMTM1.EQ.MBI) MVIM1 = MVIM1+(MV(IM2)-MVIM1)/1000.
      LER(2) = 7
C     BLCD (VIA ROOT) CALL NO. 7
      IF(ST.EQ.1.AND.IMT.NE.MBI)CALL ROOT(MVIM1,MV(IMT),TH,BL1,
     1   DTLR,ANS,AAA)
      LER(2) = 8
C     BLCD (VIA ROOT) CALL NO. 8
      IF(ST.EQ.2)CALL ROOT(MVIM1,MV(IMT),TH,BL2,DTLR,ANS,AAA)
      IF(ST.EQ.1.AND.IMT.EQ.MBI) ANS = MV(MBI)
```

```
      SPM(IMT) = ANS
      USP(IMT)= BV(ST)
C
C  CALCULATE RHO*W-SUB-THETA AND THEN RHO*W AND BETA IN THE REGION
C
  150 NSP= IMT-IM1+1
      CALL SPLINE(SPM(IM1),USP(IM1),NSP,DUDM(IM1),AAA(IM1))
      FIRST=1
      IF (IM1.NE.1) FIRST=IM2
      LAST= MM
      IF (IMT.NE.MM) LAST=IMTM1
      IF (FIRST.GT.LAST) GO TO 170
      DO 160 I=FIRST,LAST
      RWT = -DUDM(I)*WTFL/BE(I)
      IP = IPF(I,IT)
      W(IP) = SQRT(RWT**2+RWM(IP)**2)
  160 BETA(IP) = ATAN2(RWT,RWM(IP))*57.295779
C
C  CALCULATE RHO*W ON THE BLADE SURFACES
C
  170 IF (IM1.EQ.1) GO TO 180
      CALL SEARCH (SPM(IM1),S1,IHS)
      ANS = -DUDM(IM1)*WTFL/BEH(IHS,S1)
      WTB(IHS,S1) = ABS(ANS)*SQRT(1.+1./(RMH(IHS,S1)*DTDMH(IHS,S1))**2)
  180 IF(IMT.EQ.MM) GO TO 200
      CALL SEARCH (SPM(IMT),ST,IHS)
      ANS = -DUDM(IMT)*WTFL/BEH(IHS,ST)
      WTB(IHS,ST) = ABS(ANS)*SQRT(1.+1./(RMH(IHS,ST)*DTDMH(IHS,ST))**2)
  190 GO TO 20
  200 IT = IT+1
      GO TO 10
      END
```

```
$IBFTC SEARCH  DEBUG

      SUBROUTINE SEARCH (DIST,SURF,IS)
C
C  SEARCH LOCATES THE POSITION OF A GIVEN VALUE OF M IN THE MH ARRAY
C
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DO 10 I=1,100
      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
```

```
$IBFTC VELOCY  DEBUG
      SUBROUTINE VELOCY
C
C  VELOCY CALLS SUBROUTINES TO CALCULATE DENSITIES AND VELOCITIES
C  THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND IT PLOTS
C  THE SURFACE VELOCITIES
C
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      DIMENSION KKK(18)
      DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
     1   XDOWN(400),YACROS(400)
      EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
     1   (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
     2   (K(1),YACROS(1))
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA KKK(4)/1H*/,KKK(6)/1HO/,KKK(8)/1H+/,KKK(10)/1HX/
C
C  CALL VELP, VELBB, AND VELSUR THROUGHOUT THE REGION
C
      CALL VELP(1,MBIM1)
      CALL VELBB(MBI,MBO)
   20 CALL VELP(MBOP1,MM)
      CALL VELSUR
C
C  PREPARE INPUT ARRAYS FOR PLOT OF VELOCITIES
C
      IF(SURVL.LE.0) RETURN
      NP2 = 0
C    TANGENTIAL COMPONENTS
      DO 50 SURF=1,2
      NP1 = NP2
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 40
      DO 30 IHS=1,IMSS
      IF (ABS(DTDMH(IHS,SURF)*RMH(IHS,SURF)).LT..57735) GO TO 30
      NP1 = NP1+1
      YACROS(NP1) = WTB(IHS,SURF)
      XDOWN(NP1) = MH(IHS,SURF)
   30 CONTINUE
   40 KKK(2*SURF+1) = NP1-NP2
```

```
   50 NP2 = NP1
C    MERIDIONAL COMPONENTS
      DO 80 SURF=1,2
      NP1 = NP2
      DO 60 IM=MBIP1,MBOM1
      IF (ABS(DTDMV(IM,SURF)*RM(IM)).GT.1.7321) GO TO 60
      NP1 = NP1+1
      YACROS(NP1) = WMB(IM,SURF)
      XDOWN(NP1) = MV(IM)
   60 CONTINUE
   70 KKK(2*SURF+5) = NP1-NP2
   80 NP2 = NP1
C
C   PLOT VELOCITIES
C
      KKK(1) = 1
      KKK(2) = 4
      P = 5.
      WRITE(6,1000)
      CALL PLOTMY(XDOWN,YACROS,KKK,P)
      WRITE(6,1010)
      RETURN
 1000 FORMAT(2HPT,50X,24HBLADE SURFACE VELOCITIES)
 1010 FORMAT (2HPL,37X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
     1(M) DOWN THE PAGE  /2HPL/
     2   2HPL,50X,50H+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT/
     3   2HPL,50X,50H* - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT/
     4   2HPL,50X,50HX - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT/
     5   2HPL,50X,50HO - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT)
      END
```

```
$IBFTC VEL      DEBUG
      SUBROUTINE VEL
C
C  VEL CALCULATES DENSITIES AND VELOCITIES FROM THE PRODUCT OF
C  DENSITY TIMES VELOCITY
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /RHOS/RHOHB(100,2),RHOVB(100,2)
      DIMENSION WWCRM(100,2),WWCRT(100,2),SURFL(100,2)
      DIMENSION W(2500),RWM(2500),BETA(2500),WMB(100,2),WTB(100,2),
     1   XDOWN(400),YACROS(400)
      EQUIVALENCE (A(1,1),W(1)),(A(1,2),RWM(1)),(A(1,3),BETA(1)),
     1   (A(1,4),WMB(1)),(A(201,4),WTB(1)),(A(401,4),XDOWN(1)),
     2   (K(1),YACROS(1))
C
C  VELP CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT
C  INTERSECT BLADES
C
      ENTRY VELP(FIRST,LAST)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      IF(FIRST.GT.LAST) RETURN
      IF (FIRST.EQ.1.AND.INTVL.GT.0) WRITE(6,1000)
      IF (FIRST.EQ.1) RELER = .0
      DO 20 IM=FIRST,LAST
      IPU = IV(IM)
      IPL = IPU+NBBI-1
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RM(IM))**2
      LER(1)=4
      DO 10 IP=IPU,IPL
C     DENSTY CALL NO. 4
      CALL DENSTY(W(IP),RHO(IP),ANS,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
   10 W(IP) = ANS
      IF (INTVL.LE.0) GO TO 20
      WRITE (6,1010) IM,(W(IP),BETA(IP),IP=IPU,IPL)
   20 CONTINUE
      RETURN
C
C  VELBB CALCULATES ALONG VERTICAL MESH LINES WHICH INTERSECT BLADES
C
```

```
      ENTRY VELBB(FIRST,LAST)
      IF(FIRST.GT.LAST) RETURN
      IF (FIRST.NE.MBI) GO TO 30
      SURFL(MBI,1) = 0.
      SURFL(MBI,2) = 0.
   30 DO 70 IM=FIRST,LAST
      ITVU = ITV(IM,1)
      ITVL = ITV(IM,2)
      IPUP1 = IPF(IM,ITVU)
      IPLM1 = IPF(IM,ITVL)
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RM(IM))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      IF (ITVL.LT.ITVU) GO TO 50
C  ALONG THE LINE BETWEEN BLADES
      LER(1)=5
      DO 40 IP=IPUP1,IPLM1
C     DENSTY CALL NO. 5
      CALL DENSTY(W(IP),RHO(IP),ANS,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
   40 W(IP) = ANS
      IF (INTVL.LE.0) GO TO 50
      WRITE (6,1010) IM,(W(IP),BETA(IP),IP=IPUP1,IPLM1)
C  ON THE UPPER SURFACE
   50 RHOB = RHOVB(IM,1)
      LER(1)=6
C     DENSTY CALL NO. 6
      CALL DENSTY(WMB(IM,1),RHOVB(IM,1),ANS,TWLMR,CPTIP,EXPON,RHOIP,
     1   GAM,AR,TIP)
      WMB(IM,1) = ANS
      WWCRM(IM,1) = WMB(IM,1)/WCR
      IF(IM.EQ.MBI) GO TO 60
      DELTV = TV(IM-1,1)-TV(IM,1)
      SURFL(IM,1) = SURFL(IM-1,1)+SQRT((MV(IM)-MV(IM-1))**2+
     1   (DELTV*(RM(IM)+RM(IM-1))/2.)**2)
   60 RELER = AMAX1(RELER,ABS((RHOB-RHOVB(IM,1))/RHOVB(IM,1)))
C  ON THE LOWER SURFACE
      RHOB = RHOVB(IM,2)
      LER(1)=7
C     DENSTY CALL NO. 7
      CALL DENSTY(WMB(IM,2),RHOVB(IM,2),ANS,TWLMR,CPTIP,EXPON,RHOIP,
     1   GAM,AR,TIP)
      WMB(IM,2) = ANS
      WWCRM(IM,2) = WMB(IM,2)/WCR
      IF(IM.EQ.MBI) GO TO 70
      DELTV = TV(IM-1,2)-TV(IM,2)
      SURFL(IM,2) = SURFL(IM-1,2)+SQRT((MV(IM)-MV(IM-1))**2+
     1   (DELTV*(RM(IM)+RM(IM-1))/2.)**2)
   70 RELER = AMAX1(RELER,ABS((RHOB-RHOVB(IM,2))/RHOVB(IM,2)))
      RETURN
C
C  VELSUR CALCULATES ALONG A BLADE SURFACE
C
      ENTRY VELSUR
      DO 90 SURF=1,2
      IMSS = IMS(SURF)
      IF(IMSS.EQ.0) GO TO 90
      DO 80 IHS=1,IMSS
```

```
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RMH(IHS,SURF))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHOB = RHOHB(IHS,SURF)
      LER(1)=8
C     DENSTY CALL NO. 8
      CALL DENSTY(WTB(IHS,SURF),RHOHB(IHS,SURF),ANS,TWLMR,CPTIP,
     1   EXPON,RHOIP,GAM,AR,TIP)
      WTB(IHS,SURF) = ANS
      WWCRT(IHS,SURF) = WTB(IHS,SURF)/WCR
   80 RELER = AMAX1(RELER,ABS((RHOB-RHOHB(IHS,SURF))/RHOHB(IHS,SURF)))
   90 CONTINUE
      IF (RELER.LT..001) IEND = IEND+1
      WRITE(6,1080) ITER,RELER
C
C   WRITE ALL BLADE SURFACE VELOCITIES
C
      IF (SURVL.LE.0) RETURN
      WRITE(6,1020)
      WRITE(6,1040) (MV(IM),WMB(IM,1),BETAV(IM,1),SURFL(IM,1),
     1   WWCRM(IM,1),WMB(IM,2),BETAV(IM,2),SURFL(IM,2),WWCRM(IM,2),
     2   IM=MBI,MBO)
      WRITE(6,1050)
      DO 100 SURF=1,2
      IMSS = IMS(SURF)
      IF(IMSS.LT.1) GO TO 100
      WRITE(6,1060) SURF
      WRITE(6,1070) (MH(IHS,SURF),WTB(IHS,SURF),BETAH(IHS,SURF),
     1   WWCRT(IHS,SURF), IHS=1,IMSS)
  100 CONTINUE
      RETURN
 1000 FORMAT(1H1/40X,34HVELOCITIES AT INTERIOR MESH POINTS )
 1010 FORMAT(1HL,3HIM=,I3,5(24H    VELOCITY    ANGLE(DEG))/
     1(5X,5(G15.4,F9.2)))
 1020 FORMAT(1H1/16X,1H*,18X,49HSURFACE VELOCITIES BASED ON MERIDIONAL C
     1OMPONENTS,40X,1H*/16X,1H*,53X,1H*,53X,1H*/16X,1H*,19X,15HBLADE SUR
     2FACE 1,19X,1H*,20X,15HBLADE SURFACE 2,18X,1H*/7X,1HM,8X,1H*,2(3X,
     38HVELOCITY,3X,23HANGLE(DEG) SURF. LENGTH,5X,5HW/WCR,6X,1H*))
 1040 FORMAT((1H ,G13.4,3H   *,2(G12.4,F9.2,2G15.4,3H   *)))
 1050 FORMAT(1H1/3X,49HSURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
     1   )
 1060 FORMAT(//22X,15HBLADE SURFACE   ,I1/7X,1HM,10X,8HVELOCITY,3X,10HANG
     1LE(DEG),3X,5HW/WCR)
 1070 FORMAT(1H ,2G13.4,F9.2,G15.4)
 1080 FORMAT(14HLITERATION NO.,I3,3X,36HMAXIMUM RELATIVE CHANGE IN DENSI
     1TY =,G11.4)
      END
```

```
$IBFTC SPLINE  DEBUG
      SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
C
C  SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C  END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
C  ADJACENT POINT
C
      COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
     1   G(200)
      DIMENSION X(N),Y(N),EM(N),SLOPE(N)
      INTEGER Q
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-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(1)=1.
      B(N)=1.
      C(1) = -.5
      F(1)=0.
      F(N)=0.
      W(1)=B(1)
      SB(1)=C(1)/W(1)
      G(1)=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(1)=-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,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
     12HEM/(4F20.8))
      END
```

```
$IBFTC SPLINT  DEBUG
      SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT,DYDX)
C
C  SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
C  FOR A SPLINE CURVE
C  END CONDITION - SECOND DERIVATIVES ARE THE SAME AT END POINT AND
C  ADJACENT POINT
C
      COMMON Q/BOX/S(100),A(100),B(100),C(100),F(100),W(100),SB(100),
     1   G(100),EM(100)
      DIMENSION   X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX)
      INTEGER Q
      IF(MAX.LE.0) RETURN
      III = 0
      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(1)=1.0
      B(N)=1.0
      C(1) = -.5
      F(1)=0.0
      F(N)=0.0
      W(1)=B(1)
      SB(1)=C(1)/W(1)
      G(1)=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(1)) 70,60,90
   60 YINT(I)=Y(1)
      GO TO 130
   70 IF(Z(I).GE.(1.1*X(1)-.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)+(Y(K)/S(K)-EM(K)*S(K)/6.)*(Z(I)-X(K-1))+(Y(K-1)/S(K)-EM(K-1)
     2*S(K)/6.)*(X(K)-Z(I))
  130 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/S(K)+EM(K)*(X(K-1)-Z(I))**2/2.
     10/S(K)+(Y(K)-Y(K-1))/S(K)-(EM(K)-EM(K-1))*S(K)/6.0
  140 CONTINUE
      MXA = 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,21HNO. OF POINTS GIVEN =,I3,30H, NO. OF INTERPOLATED PO
     1INTS =,I3/10X,1HX,19X,1HY,16X,11HX-INTERPOL.,9X,11HY-INTERPOL.,
     28X,14HDYDX-INTERPOL./(5E20.8))
      END
```

```
$IBFTC SPLN22  DEBUG

      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),
     1   G(200)
      DIMENSION X(N),Y(N),EM(N),SLOPE(N)
      INTEGER Q
      DO 10 I=2,N
   10 S(I)=X(I)-X(I-1)
      NO=N-1
      IF(NO.LT.2) GO TO 30
      DO 20 I=2,NO
      A(I)=S(I)/6.
      B(I)=(S(I)+S(I+1))/3.
      C(I)=S(I+1)/6.
   20 F(I)=(Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
   30 A(N) = S(N)/6.
      B(1)=S(2)/3.
      B(N) = S(N)/3.
      C(1)=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(1)=-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,15HNO. OF POINTS =,I3/10X,1HX,19X,1HY,19X,5HSLOPE,15X,
     12HEM/(4F20.8))
      END
```

```
$IBFTC ROOT    DEBUG
      SUBROUTINE ROOT(A,B,Y,FUNCT,TOLERY,X,DFX)
C
C  ROOT FINDS A ROOT FOR (FUNCT MINUS Y) IN THE INTERVAL (A,B)
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      INTEGER SRW
      IF (SRW.EQ.21) WRITE(6,1000) A,B,Y,TOLERY
      TOLERX= (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+TOLERX*FLOAT(I)
      IF(I.EQ.1) X = B
      GO TO 10
   20 IF (X.LT.AB2) X=X+TOLERX
      IF (X.GE.AB2) X=X-TOLERX
      GO TO 10
   30 WRITE(6,1020) LER(2),A,B,Y
      STOP
 1000 FORMAT (32H1INPUT ARGUMENTS FOR ROOT -- A =G13.5,3X,3HB =,G13.5,
     1   3X,3HY =,G13.5,3X,8HTOLERY =,G13.5/17H  ITER. NO.     X,17X,
     2   2HFX,15X,3HDFX,10X,3HINF)
 1010 FORMAT (5X,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
```

```
$IBFTC BLCD    DEBUG
      SUBROUTINE BLCD
C
C  BLCD CALCULATES BLADE THETA COORDINATE AS A FUNCTION OF M
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      COMMON /INP/GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   MBI,MBO,MM,NBBI,NBL,NRSP,MR(50),RMSP(50),BESP(50),
     2   BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      COMMON /GEOMIN/ CHORD(2),STGR(2),MLE(2),THLE(2),RMI(2),RMO(2),
     1   RI(2),RO(2),BETI(2),BETO(2),NSPI(2),MSP(50,2),THSP(50,2)
      COMMON /BLCDCM/ EM(50,2),INIT(2)
      ENTRY  BL1(M,THETA,DTDM,INF)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST,SRW
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      REAL M,MMLE,MSPMM,MMMSP
      SURF= 1
      SIGN= 1.
      GO TO 10
      ENTRY  BL2(M,THETA,DTDM,INF)
      SURF= 2
      SIGN=-1.
   10 INF= 0
      NSP= NSPI(SURF)
      IF (INIT(SURF).EQ.13) GO TO 30
      INIT(SURF)= 13
C
C  INITIAL CALCULATION OF FIRST AND LAST SPLINE POINTS ON BLADE
C
      AA = BETI(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,
     1 AAA,EM(1,SURF))
      IF(BLDAT.LE.0) GO TO 30
      IF (SURF.EQ.1) WRITE(6,1000)
      WRITE(6,1010) SURF
      WRITE (6,1020) (MSP(IA,SURF),THSP(IA,SURF),AAA(IA),EM(IA,SURF),
     1   IA=1,NSP)
C
C  BLADE COORDINATE CALCULATION
C
```

```
   30 KK = 2
      IF (M.GT.MSP(1,SURF)) GO TO 50
C
C  AT LEADING EDGE RADIUS
C
      MMLE= M-MLE(SURF)
      IF (MMLE.LT.-DMLR) GO TO 90
      MMLE= AMAX1(0.,MMLE)
      THETA= SQRT(MMLE*(2.*RI(SURF)-MMLE))*SIGN
      IF (THETA.EQ.0.) GO TO 40
      RMM= RI(SURF)-MMLE
      DTDM= RMM/THETA/RMI(SURF)
      THETA= THETA/RMI(SURF)+THLE(SURF)
      RETURN
   40 INF= 1
      DTDM = 1.E10*SIGN
      THETA= THLE(SURF)
      RETURN
C
C  ALONG SPLINE CURVE
C
   50 IF (M.LE.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)
      MSPMM= MSP(KK,SURF)-M
      MMMSP= M-MSP(KK-1,SURF)
      THK= THSP(KK,SURF)/S
      THKM1= THSP(KK-1,SURF)/S
      THETA= EMKM1*MSPMM**3/6./S + EMK*MMMSP**3/6./S + (THK-EMK*S/6.)*
     1 MMMSP + (THKM1-EMKM1*S/6.)*MSPMM
      DTDM= -EMKM1*MSPMM**2/2./S + EMK*MMMSP**2/2./S + THK-THKM1-(EMK-
     1 EMKM1)*S/6.
      RETURN
C
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.E10*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,33HBLADE DATA AT INPUT SPLINE POINTS)
 1010 FORMAT(1HL,17X,16HBLADE     SURFACE,I4)
 1020 FORMAT (7X ,1HM,10X,5HTHETA,10X,10HDERIVATIVE,5X,10H2ND DERIV. /
     1  (4G15.5) )
 1030 FORMAT (14HLBLCD CALL NO.,I3/33H M COORDINATE IS NOT WITHIN BLADE/
     14H M =,G14.6,10X,6HSURF =,G14.6)
      END
```

```
$IBFTC DENSTY  DEBUG
      SUBROUTINE DENSTY(RHOW,RHO,VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP)
C
C  DENSTY CALCULATES DENSITY AND VELOCITY FROM THE WEIGHT FLOW PARAMETER
C  DENSITY TIMES VELOCITY
C
      COMMON SRW,ITER,IEND,LER(2),NER(2)
      VEL = RHOW/RHO
      IF (VEL.NE.0.) GO TO 10
      RHO = RHOIP
      RETURN
   10 TTIP = 1.-(VEL**2+TWLMR)/CPTIP
      IF(TTIP.LT.0.) GO TO 30
      TEMP = TTIP**(EXPON-1.)
      RHOT = RHOIP*TEMP*TTIP
      RHOWP = -VEL**2/GAM*RHOIP/AR*TEMP/TIP+RHOT
      IF(RHOWP.LE.0.) GO TO 30
      VELNEW = VEL+(RHOW-RHOT*VEL)/RHOWP
      IF(ABS(VELNEW-VEL)/VELNEW.LT..0001) GO TO 20
      VEL = VELNEW
      GO TO 10
   20 VEL = VELNEW
      RHO = RHOW/VEL
      RETURN
   30 TGROG = 2.*GAM*AR/(GAM+1.)
      VEL = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
      RWMORW = RHOW/RHO/VEL
      NER(1) = NER(1)+1
      WRITE(6,1000) LER(1),NER(1),RWMORW
      IF(NER(1).EQ.50) STOP
      RETURN
 1000 FORMAT(16HLDENSTY CALL NO.,I3/9H NER(1) =,I3/10H RHO*W IS ,F7.4,
     134H TIMES THE MAXIMUM VALUE FOR RHO*W)
      END
```

```
$IBFTC IPF      DEBUG
      FUNCTION IPF(IM,IT)
      COMMON /CALCON/MBIM1,MBIP1,MBOM1,MBOP1,MMM1,HM1,HT,DTLR,DMLR,
     1   PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,LAMBDA,TWL,ITMIN,ITMAX,
     2   NIP,IMS(2),BV(2),MV(100),IV(101),ITV(100,2),TV(100,2),
     3   DTDMV(100,2),BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),
     4   RMH(100,2),BEH(100,2),RM(100),BE(100),DBDM(100), SAL(100),
     5   AAA(100)
      IPF = IV(IM)+IT-ITV(IM,1)
      RETURN
      END
```

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

REFERENCES

1. Katsanis, Theodore: Computer Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-4525, 1968.
2. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Tandem Blade Turbomachine. NASA TN D-5044, 1969.
3. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities in a Magnified Region of a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-5091, 1969.
4. Mechtly, E. A.: The International System of Units. Physical Constants and Conversion Factors. NASA SP-7012, 1964.
5. Walsh, J. L.; Ahlberg, J. H.; and Nilson, E. N.: Best Approximation Properties of the Spline Fit. J. Math. Mech., vol. 11, no. 2, 1962, pp. 225-234.

NASA-Langley, 1969 —— 1 E-4706